Medium Access Control in Energy Harvesting - Wireless Sensor Networks

Fafoutis, Xenofon

Publication date:
2014

Document Version
Publisher's PDF, also known as Version of record

Medium Access Control in Energy Harvesting - Wireless Sensor Networks

Xenofon Fafoutis

Ph.D. Dissertation
Technical University of Denmark
2014

Medium Access Control in Energy Harvesting - Wireless Sensor Networks

Xenofon Fafoutis

Supervisors:
Nicola Dragoni, Associate Professor
Jan Madsen, Professor

Xenofon Fafoutis
PHD-2014-328

Technical University of Denmark
Department of Applied Mathematics and Computer Science
Embedded Systems Engineering Section
DK-2800 Kgs. Lyngby, Denmark
http://www.compute.dtu.dk/

Abstract

Focusing on Wireless Sensor Networks (WSNs) that are powered by energy harvesting, this dissertation studies energy-efficient communication links between senders and receivers that are alternating between active and sleeping states of operation. In particular, the focus lies on Medium Access Control (MAC) protocols that are following the receiver-initiated paradigm of asynchronous communication. According to the receiver-initiated paradigm the communication is initiated by the receiver that states its availability to receive data through beacons. The sender is passively listening to the channel until it receives the beacon of interest.

In this context, the dissertation begins with an in-depth survey of all the receiver-initiated MAC protocols and presents their unique optimization features, which deal with several challenges of the link layer such as mitigation of the energy consumption, collision avoidance, provision of Quality of Service (QoS) and security. Focusing on the particular requirements of an energy harvesting application, the dissertation continues with the presentation of a MAC protocol, named On Demand MAC (ODMAC), which extends the receiver-initiated paradigm with several energy-efficient features that aim to adapt the consumed energy to match the harvested energy, distribute the load with respect to the harvested energy, decrease the overhead of the communication, address the requirements for collision avoidance, prioritize urgent traffic and secure the system against beacon replay attacks.

The performance and behavior of ODMAC and its features are compared to the state-of-the-art and evaluated using mathematical models, simulations and testbed experiments that are based on eZ430-rf2500 wireless development platform. The results validate the efficient use of the harvested energy and demonstrate sustainable operation.

Abstrakt

Denne afhandling fokuserer på energieffektiv kommunikation i trådløse sensornetværk (eng. Wireless Sensor Networks), der er drevet af energi høstet fra omgivelserne. Afhandlingen beskriver energieffektive forbindelser mellem sendere og modtagere, der periodisk skifter mellem aktiv og sovende tilstande. Der er særlig fokus på Medium Access Control (MAC) protokoller, der følger det modtager-initieret paradigme med asynkron kommunikation. Ifølge dette paradigme initieres kommunikationen af modtageren gennem udsendelse af beacons, der udtrykker modtagerens tilgængelighed til at modtage data. Afsenderen lytter passivt til kanalen, indtil den modtager et beacon af interesse.

I denne sammenhæng begynder afhandlingen med en grundig undersøgelse af alle modtager-initierede MAC-protokoller og præsenterer deres unikke optimerings funktioner. Disse funktioner beskæftiger sig med flere udfordringer i linket lag, såsom minimering af energiforbruget, undgåelse af kollisioner, tilvejebringelse af Quality of Service (QoS) og sikkerhed. Med fokus på de særlige krav der stilles til applikation baseret på energi høstning, fortsætter afhandlingen med en præsentation af en ny MAC-protokol, kaldet On Demand MAC (ODMAC), som udbygger det modtager-initierede paradigme med flere energieffektive funktioner, der har til formål at tilpasse den forbrugte energi med den høstede energi, at distribuere belastningen med hensyn til den høstede energi, at mindske overhead af kommunikationen, at undgå kollisioner, at prioritere prioriteret trafik og at sikre systemet mod beacon replay angreb.

Performance og opførsel af ODMAC og dens funktioner, sammenlignes med state - of- the-art og evalueres ved hjælp af matematiske modeller, simuleringer og eksperimenter, der er baseret på den trådløse platform eZ430-rf2500. Resultaterne fra disse eksperimenter, validerer en effektiv udnyttelse af den høstede energi og demonstrerer bæredygtig drift.

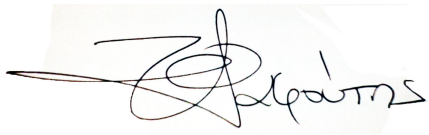
Preface

This dissertation was prepared in the Embedded Systems Engineering Section of the Department of Applied Mathematics and Computer Science of the Technical University of Denmark in fulfillment of the requirements of the PhD program.

This work was supervised by Associate Professor Nicola Dragoni and Professor Jan Madsen and conducted occasionally in collaboration with Alessio Di Mauro, Madava D. Vithanage, Charalampos Orfanidis and Associate Professor Sebastian Mödersheim.

This dissertation contains no material which has been accepted for the award of any other degree or diploma in my name, in any university or other institution and, to the best of my knowledge and belief, contains no material previously published by another person, except where due reference has been made in the text.

Kgs. Lyngby, 31/1/2014



Xenofon Fafoutis

Acknowledgements

The completion of this dissertation marks the end of long journey and I wish to express my gratitude to a number of persons who provided me with their help and support over the years, in a direct or indirect way.

Starting with my supervisors, Nicola and Jan, I would like to thank you for selecting me and trusting me with this PhD project in the first place. A special thanks to Nicola for providing me with a working environment of peace and serenity that is fundamental for research ideas to emerge and develop and for supporting me and allowing me to pursue my ideas and lead my research towards what I was finding most interesting. A special thanks to Jan for helping me translate the abstract in Danish and for bringing me in contact with Thomas and WindowMaster and trusting me with their project. I am really grateful for this experience. I wish I will have the opportunity in the future to collaborate with both of you.

I would also like to thank the members of my examination committee, Professors Alberto Nannarelli, Juha Petteri Plosila and Jüri Vain. I am grateful for the time you spent reading my dissertation, your contributions to its improvement and the discussion that we had during the PhD defence. A special thanks to Alberto for providing me with lab equipment and commenting on my papers.

A very special thanks to my M.Sc. supervisor in the University of Crete and ICS-FORTH, Prof. Vasilios Siris, for teaching me how to do research. Without you, this dissertation would not have been the same. I wish we will find an opportunity to collaborate again in the future.

My special thanks to Alessio for introducing me to the world of DIY electronics. I will

never forget the countless hours we spent trying to make things work, fighting against the evil forces of black magic, and the feelings of satisfaction when, finally, things did work. I will never forget Amber, our very first DIY audio electronics project.

My special thanks to my Master students, Madava and Haris, for our collaboration during their projects. Madava, the discussions we had, late at night in the lab, were very productive and some papers you found pushed my research forward. Haris, without your work, I would not have been able to provide such an in-depth evaluation of my protocols, within the time constraints of my PhD project.

I would like to thank Thomas Sørensen for his advice during our project with Window-Master. I will never forget the things you told me, in our meeting in DTU, on how to prioritize my work in the industrial world within very tight deadlines.

I would like to thank Prof. Vangelis Angelakis for inviting me to give a talk in his group in the University of Linköping and for inviting me to participate in the short course of Prof. Antony Ephremides. It was a really inspiring experience.

I would like to thank Karin for the warm welcome in DTU and for providing me all the assistance I could possibly need.

A big thanks to the Friday night group; Sahar, Valia, Alessio, Laura, Paolo, Domi, Alex, Giovanni and Gosia. And to the older members of the group; Massimo, Valentina, Edu, Nicola, Stavros, Alex, Davide and Seliz. Without the occasional beer (or two) and a terrible pizza, who can do do research? I am looking forward to seeing you again either in the Cellar Bar, in the 322 kitchen or anywhere else. A special thanks to Laura for hosting me in Eindhoven and to Nicola for hosting me in Padova. Man, that concert of Radiohead was really great.

My very special thanks to my old friend, Menelaos, for giving me these books on that Christmas break of 2011 and for inspiring me to start playing my guitar again. It changed my life. My very special thanks to my friend, Giorgos (aka DJ Katakoubas), for our discussion during that concert in Technopolis. When will we have the next Jims jam session? All the best with Duoyu.

Many thanks to Classy Fire; Ali, Marcin, Francesco, Dhanny, Jacob and Kasia. Our weekly jam session was fundamental for surviving and going on. Special thanks to Marcin for pushing us to give that gig in Magneten. Nobody believed in it as much as you did and, at the end, it was an amazing experience that would not have happened without you. All my wishes to Maya. Special thanks to Ali for offering to me his voice for my recordings. I wish we will find an opportunity to jam again.

Cheers to Pantelis, Nassos and Lazaros for visiting me and for the experiences we shared in Copenhagen. Cheers to Evi for Roskilde Festival. Cheers to Michael for the

gigs of Jack White and The Black Keys.

My warmest gratitude to my family; my father, my mother, my grandfather, my grandmother and my sister, Katerina. Thanks for your love. Thanks for supporting me in whatever decision I have ever taken in my life, no matter how crazy does it sound. Thanks for being there whenever I need you. I love you all.

My warmest gratitude to Letizia. Thank you for the beach day. Thank you for the concert in Bilbao. Thank you for the pillow idea. Thank you for that basketball match. Thank you for your trust after the Irish pub. Thank you for the vinyl record. Thank you for the sandwich on the bench in the night of the half-woman half-shrimp. Thank you for your stories. Thank you for your jokes. Thank you for that first restaurant and that walk around Magdalena. Thank you for the car trips. Thank you for sacrificing your sleep in Vitoria. Thank you for your gift-giving madness. Thank you for the key chain. Thank you for the gifts with the cards. Thank you for the cake in my doorstep. Thank you for that little bird that makes this funny sound. Thank you for the lion. Thank you for liking my CV. Thank you for the Barcelona paper. Thank you for that football match. Thank you for the white brown sugar. Thank you for the chupa chups. Thank you for the pizza. Thank you for the cafetiera. Thank you for repeating when I forget. Thank you for all these long moments. Thank you for your acceptance. Thank you for your tolerance. Thank you for the PhD defence.

Contents

List of Figures

List of Tables

Abbreviations

AB Altruistic Backoff. 23, 24, 43, 54, 58, 64, 89–91, 93–96, 123, 130, 131, 134, 141, 145–149, 172–174

ABR Altruistic Backoff Request. 58, 59, 91, 93, 96, 134, 138

ACK Acknowledgment. 33, 34, 36, 46, 54

ADB Asynchronous Duty cycle Broadcasting. 45, 46, 50–52

ADC Analog-to-Digital Converter. 139, 141, 142, 153, 166

AIF Application Integration Framework. 102

AMIs Advanced Metering Infrastructures. 120, 122

AMR Automatic Meter Reading. 120, 121

AnB Alice-and-Bob. 100, 101

ANC Announcement. 46

AP Access Point. 152–154, 160

APP Application. 78, 80, 81

APs Access Points. 3

ARM Asynchronous Receiver-initiated Multi-channel MAC. 47

ARP Address Resolution Protocol. 153, 159

ATPC Adaptive Transmission Power Control. 14

AVISPA Automated Validation of Internet Security Protocols and Applications. 100

B-MAC Berkley MAC. 21, 110

BANs Body Area Networks. 6

BC Broadcast Channel. 46, 47

BCN Beaconing Scheme. 113

BEB Binary Exponential Backoff. 34, 49, 79, 90, 91, 93

BER Bit Error Rate. 17

CA Collision Avoidance. 30, 31

CAP Contention Access Period. 20

CB Constant Backoff. 91, 93, 94, 134, 145–148

CBC Cipher-Block Chaining. 62, 103, 138

CC Control Channel. 46

CCA Clear Channel Assessment. 34, 37, 40, 47, 54, 80, 124, 130, 134

CCB Command & Control Beacon. 62

CFP Contention Free Period. 20

CO2 Carbon Dioxide. 23, 24, 151–153, 155, 159, 175

CRCs Cyclic Redundancy Checks. 29

CSMA Carrier Sense Multiple Access. 18, 19

CSMA/CA Carrier Sense Multiple Access with Collision Avoidance. 18–20, 152

CTS Clear-to-Send. 35, 46

CW Contention Window. 58, 90, 91

CyMAC Delay Bounded MAC. 44, 49–52

DC Duty Cycle. 17, 143

DCF Distributed Coordination Function. 18, 90, 152

DCM Duty Cycle Multi-channel MAC. 46, 47, 49, 51, 52

DCs Duty Cycles. 18

DEHAR Distributed Energy Harvesting Aware Routing. 16, 17

DHCP Dynamic Host Configuration Protocol. 153, 159

DoS Denial-of-Service. 14, 59, 60, 98, 174

ECO Energy Consumption Overhead. 105

EE-RI-MAC Energy Efficient RI-MAC. 37, 49, 51, 52

EH-AMIs Energy Harvesting - Advanced Metering Infrastructures. 120

EH-HCA Energy Harvesting - Heat Cost Allocator. 120, 121

EH-WSN Energy Harvesting - Wireless Sensor Network. 72, 75, 127

EH-WSNs Energy Harvesting - Wireless Sensor Networks. 7, 9, 15–18, 22, 23, 42, 53, 54, 65, 109, 173

EM-MAC Efficient Multi-channel MAC. 47–49, 51, 52

ENO Energy Neutral Operation. 9, 10, 19, 42, 55, 71, 75, 77, 84

FFDs Full Function Devices. 20

FR Frame Reordering. 41

FSM Finite State Machine. 132

GPRS General Packet Radio Service. 121

GPS Global Positioning System. 11, 12

GREES Geographic Routing with Environmental Energy Supply. 17

GTS Guaranteed Time Slot. 20

HCA Heat Cost Allocator. 120

HCAs Heat Cost Allocators. 120, 121

HCR Harvested-to-Consumed power Ratio. 75–77, 84

HTTP Hypertext Transfer Protocol. 154, 155, 159

IEEE Institute of Electrical and Electronic Engineers. 18, 20, 23, 90, 151, 152, 160, 175

IMR+ Inter-Meter Reading +. 24, 120–123, 126–130, 174

IP Internet Protocol. 153

IRDT Intermittent Receiver-driven Data Transmission. 36, 50–52

LAR Layer-based Anycast Routing. 63, 66, 73, 78, 79, 112, 134, 172

LBM Loose Binding Mode. 62, 71, 75

LEACH Low Energy Adaptive Clustering Hierarchy. 15, 16, 99

LED Light Emitting Diode. 141

LMA Local Mean Algorithm. 14

LMN Local Mean of Neighbors Algorithm. 14

LPM3 Low Power Mode 3. 133

LPP Low Power Probing. 31, 38

LR-WPANs Low Rate - Wireless Personal Area Networks. 20

LST Low Surface Temperature. 8, 120

MAC Medium Access Control. i, iii, 18–24, 27–31, 33, 36, 37, 39, 46–50, 53–58, 60, 62–64, 68, 69, 78, 79, 90, 97–99, 109, 110, 114, 116, 118, 120, 122, 126, 129, 130, 152, 153, 171–175

MCU Microcontroller Unit. 132, 133, 138–142, 145, 152, 153

MP Minimum Path. 16, 17

NOCA No Collision Avoidance. 134

OC-MAC Opportunistic Cooperation MAC. 35, 49, 51, 52

ODMAC On Demand MAC. i, iii, 23, 24, 28, 42, 43, 48, 49, 51–55, 57, 60, 62–65, 72, 76, 78–82, 84, 86, 89, 97–99, 109–113, 120, 122, 124–132, 134, 140, 146, 160, 168, 171–175

OFMC On-the-Fly Model Checker. 100, 102

OSI Open Systems Interconnection. 10

PHY Physical. 20, 79–81

PRE Preamble Scheme. 113

PV Photo-Voltaic. 141, 143

PW-MAC Predictive Wake-up MAC. 40, 41, 49, 51, 52

QAEE-MAC QoS Aware Energy-Efficient MAC. 44, 50–52

QoS Quality of Service. i, iii, 30, 43, 54, 59, 94, 173, 174

R-MPE Randomized Minimum Path Energy. 16, 17

R-MPRT Randomized Minimum Path Recovery Time. 16, 17

R-WMP Randomized Weighted Minimum Path. 16, 17

RAP Receiver Authentication Protocol. 23, 24, 28, 48, 50–52, 54, 59–61, 64, 97, 99–104, 106, 173, 174

RAP-D Receiver Authentication Protocol - Detection. 48, 60, 61, 100, 102, 104, 106, 108

RAP-P Receiver Authentication Protocol - Prevention. 48, 60, 61, 102, 105, 106

RB Random Backoff. 46, 58, 59, 89–91, 93, 145

RC-MAC Receiver-Centric MAC. 36, 49, 51, 52

REA-MAC Routing-Enhanced Asynchronous MAC. 38, 49, 51, 52

RFDs Reduced Function Devices. 20

RI-MAC Receiver Initiated MAC. 21, 28, 31, 33–37, 39–41, 44, 45, 48, 49, 51, 52, 90, 168

RICER Receiver Initiated Cycled Receiver. 21, 28, 31, 32, 51, 52

RP-MAC Reordering Passive MAC. 41, 49, 51, 52

RSSI Received Signal Strength Indicator. 12

RTR Ready-to-Receive. 33

RTS Ready-to-Send. 35, 36, 46

RW-MAC Receiver Wake-up MAC. 40, 41, 46, 49, 51, 52, 62

RWB Receiver Wake-up Broadcast. 46, 50–52

S-MAC Sensor-MAC. 20

SARI-MAC Self Adapting RI-MAC. 43, 49, 51, 52

SETW Synchronization Error Tolerance Window. 45

sLEACH Solar Low Energy Adaptive Clustering Hierarchy. 15, 17

SYNC Synchronization. 20

TCP Transmission Control Protocol. 154, 155

TCP/IP Transmission Control Protocol / Internet Protocol. 152, 153, 160

UDP User Datagram Protocol. 153–155, 159

WideMAC Wide-band MAC. 39, 49, 51, 52

WLAN Wireless Local Area Network. 3, 152

WPA2 Wi-Fi Protected Access II. 154

WPS Wi-Fi Protected Setup. 154

WSN Wireless Sensor Network. 2, 5, 6, 11, 14, 22, 23, 68, 74, 100, 121, 151, 161

WSNs Wireless Sensor Networks. i, 1, 2, 4–7, 9, 11, 13–15, 18, 20, 27, 29, 30, 46, 48, 50, 58, 59, 90, 98, 100, 110, 120, 161, 164, 167, 168

X-MAC Short Preamble MAC. 21, 24, 37, 109–113, 130, 174

YA-MAC Yet Another MAC. 45, 50–52

CHAPTER 1

Introduction

1.1 Wireless Sensor Networks

Wireless Sensor Networks (WSNs) [133] have attracted a lot of attention in the last decade in both the academic and industrial world. Recent advances in wireless technologies and microcontrollers have made possible the realization of systems of multiple networked embedded computing devices that are able to sense, measure and gather information from the environment they are deployed into. Such devices are spatially distributed in a monitored area and their goal is to cooperatively pass the collected information to a central station, also known as *sink*, for storage and analysis. In an attempt to avoid excessive usage of wires, but also due to the possibility of outdoors deployments, sensor networks depend on wireless communications for data transfer.

1.1.1 The Resources and Design Priorities of a Sensor Node

Sensor nodes are embedded devices equipped with a sensor unit, a microcontroller, a wireless radio and a power source. A sensor node, as a whole system, is constrained by the limited resources of its separate modules. The resource constraints of a sensor node can be summarized as storage, processing and energy constraints. Microcontrollers

are characterized by low memory resources and processing capabilities, while a sensor node is, typically, powered by batteries.

The greatest challenge faced in the field of WSNs lies in the energy consumption of a sensor node. The usefulness of a sensor expires when its battery runs out. Due to large or unaccessible deployments, battery replacement substantially increases the cost of network maintenance. Therefore, the design of wireless sensor systems and protocols for WSNs is primarily based on the efficient management of the available energy, also knowns as *energy-efficiency*. Since the wireless radio is orders of magnitude more energy consuming than the microcontroller or other parts of the system (e.g. [115]), the energy-efficiency of WSNs heavily depends on the efficient management of the radio unit [7].

To achieve energy-efficiency, the development of a sensor node is based on a minimalistic design. Both the hardware and firmware, but also the network itself, are tailored to the properties of the surrounding environment and the needs of the running application. Unless required to operate, all hardware modules, inside or outside the microcontroller, are shut down or put into sleep mode. Moreover, the operating system and the networking protocols are stripped down from unnecessary features or algorithms. Any unnecessary action compromises the energy-efficiency of the system and, thus, must be avoided.

1.1.2 Other Types of Nodes

In addition to sensor nodes, a Wireless Sensor Network (WSN) contains other classes of nodes. The most important one is the sink node, whose main purpose is to gather all the sensed data from the sensor nodes for storage and analysis. A WSN is not necessarily constrained to a single sink node. Multi-sink deployments consist of multiple sink nodes and the goal of the sensor nodes is to pass the sensed information to either one of them. The sink node is assumed to be a standard computing system that is plugged into the mains power supply. Therefore, it is safe to assume that from the perspective of the WSN, a sink node has unlimited energy, memory and processing resources.

Lastly, a third type of nodes is found in WSNs. In literature, the name of such nodes may vary depending on their role. Yet, they are characterized by resource constraints that are less tight in comparison to sensor nodes. Often, such nodes take the role of cluster leaders, and become responsible for collecting data from their neighborhood and forwarding it to the sink in an hierarchical manner (see for instance [121]).

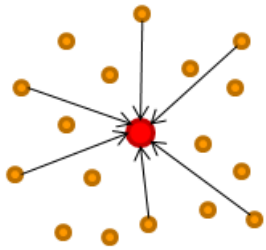


Figure 1.1: Single-hop star topology. All sensor nodes can directly communicate with the sink node.

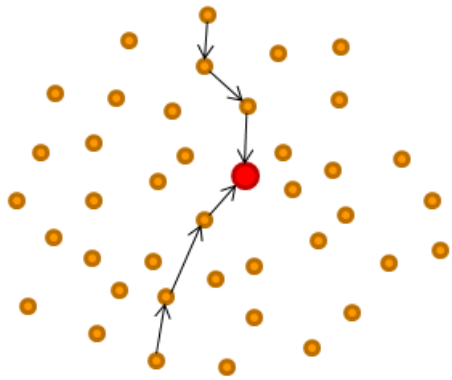


Figure 1.2: Multi-hop topology. Sensor nodes forward traffic of other sensor nodes towards the sink node.

1.1.3 Network Structure

According to how different types of nodes are structured, we can distinguish two fundamental types of topologies. The first one is single-hop star topology where sensor nodes directly communicate with a sink node (e.g. Figure 1.1). From a wireless networking point of view this topology resembles the Wireless Local Area Network (WLAN) model which consists of Access Points (APs) and mobile stations (e.g. laptops). In this topology, the sensor nodes do not have forwarding duties, i.e. they do not receive and forward data on behalf of other sensor nodes. Then, there is the multi-hop topology (e.g. Figure 1.2) where sensor nodes are deployed in an wider area than the coverage of their radio. Therefore, they have forwarding duties, as they cannot directly communicate with the sink node. A specific type of multi-hop topologies is the cluster-based

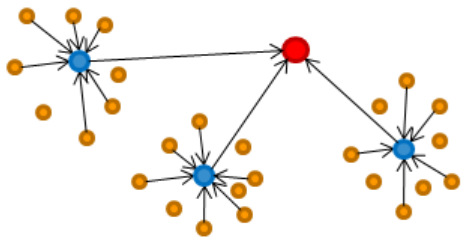


Figure 1.3: Cluster-based multi-hop topology. Sensor nodes forward traffic to cluster leaders that foward the traffic to the sink node.

multi-hop topology (e.g. Figure 1.3). In this case, cluster leaders collect the data of their neighborhood and forward it to the sink.

Mobility introduces dynamics in the networks structure. Mobile WSNs may consist of either mobile sink nodes or mobile sensor nodes. In the former case, the sink node periodically moves in close proximity to the sensor nodes and polls them for data. In the latter case, the sensor nodes are able to move and reposition themselves in the environment. Challenges of mobile nodes include optimum deployment, localization and navigation of the nodes.

1.1.4 Applications

The information that a sensor node can extract from the environment, depends on its sensor unit. A lot of different types of sensors can be attached into a sensor node, including mechanical, thermal, biological, chemical, optical and magnetic sensors. Environmental monitoring is one of the most typical applications. Sensors can measure various properties of the environment like temperature, light, barometric pressure, humidity, acidity and carbon dioxide concentration. Detecting or tracking objects, animals or humans, constitutes another major application theme, which is based on microphones, low-resolution cameras, accelerometers and other types of sensors.

WSNs can support a wide range of different applications that can be classified into two main categories, monitoring and tracking applications. Monitoring applications include environmental monitoring, industrial monitoring, factory and process automation, health monitoring and logistics storage support. Tracking applications include detecting or tracking events, objects, animals, people or vehicles. Tracking services can be useful in multiple fields such as military, businesses and public transportation networks. Hybrid applications that fall into both categories may also exist.

The two main application categories are loosely characterized by two basic traffic patterns, the continuous and the event-driven [119]. The continuous traffic pattern is the dominant traffic pattern in monitoring applications. In this case, the sensor nodes periodically sense the environment and report data to the sink. The event-driven traffic pattern is dominant in the tracking applications. In this case, the traffic is triggered and generated by an external unpredictable event of interest. Generally, tracking applications require additional continuous traffic for the purpose of negative acknowledgments. In other words, the network administrator requires a way to differentiate between the case of a non-working sensor node and a sensor node that does not detect the event of interest.

WSN applications have low requirements compared to traditional wireless networks. Depending on the nature of an application, throughput and delay requirements may exist, but are several orders of magnitude lower than traditional wireless networks. Furthermore, retransmissions may be obsolete, as they can be replaced with fresh measurements. Typically, the priority of the application requirements comes after the requirement for energy-efficiency. We distinguish two basic types of applications based on their requirements. First is *delay-sensitive* applications, where short delay is the primary performance priority. Consider, for example, a sensor network for fire detection. Then, there are application *offline-analysis* applications where the primary priority is the amount of measurements (i.e throughput). In such applications, the goal is to gather enough measurements to monitor how a phenomenon changes over time over a longer period. Consider, for instance, a sensor network for weather forecasting.

1.1.4.1 Example Applications

Some typical examples of deployed WSN are briefly presented next. For more WSN applications we refer the reader to the following survey [5].

Environmental monitoring is the dominant application of WSNs. There are both indoor and outdoor deployments. An example of an indoor deployment is presented in [22], where a set of wireless sensors were installed in U.C. Berkley to monitor the light and temperature. The capability of sensing temperature, light, status of windows and doors, air streams and indoor air pollution can be utilized for optimal control of the indoor environment. An example of an outdoor deployment is the WSN on Great Duck Island [78]. The sensor network was used to sense the temperature, barometric pressure and humidity of the environment that the birds live. The aim of the project was to monitor their behavior to climatic changes.

WSNs can be used for *military applications*, providing services such as information collection, enemy tracking, battlefield surveillance and target classification. For example, in [118] a project named ”A Line in the Sand” is presented. It refers to the

deployment of a 90-node WSN that is capable of detecting metallic objects, aiming at tracking and classifying moving objects with significant metallic content such as vehicles or armed soldiers. Other beings, such as civilians, were ignored by the sensors.

WSNs also have *animal tracking applications*, such as the study of an endangered species, the red wolf [11]. The concept was to attach a node in each wolf and record information about its condition and behavior. These mobile sensor nodes were transmitting the sensed data when the wolf passed by a static sensor node that was always connected to the WSN.

Another industrial use of WSNs is *support for logistics* for inventory control and storage management. British Petroleum (BP), in [62], describes an application of wireless sensors in warehouses supporting the storage management of barrels. The idea is that sensors attached to barrels will be able to sense nearby barrels, identify their content and issue alerts in case of content incompatibilities that might lead to an explosion.

The are also *human-centric applications*. Health science, for instance, can benefit from WSNs. Reference [77] presents how wireless sensors can support senior citizens. The sensor network can identify behaviors that indicate early stages of disorders. Wireless sensors can also be used to record actions (e.g. taking meditation), indicate behaviors that patients may hide from their doctor or detect emergencies.

Lastly, another interesting application is *wearable sensors*, also known as Body Area Networks (BANs). An example is presented in [93]. A set of six wireless sensors were attached to a glove, one at each finger and one at the wrist. The objective of this application was movement and gesture recognition. Such application can potentially be useful in many fields, such as the development of wireless wearable input devices, gesture recognition for the disabled and work training in simulated environments.

1.2 Energy Harvesting - Wireless Sensor Networks

Advances in battery technologies are not enough to cover the demands of many WSN applications. Energy-efficient system design and energy-aware communication protocols are able to provide long periods of operation, without battery recharging and replacement. However, a fundamental trade-off between energy-efficiency and performance arises. Essentially, WSNs need to find the perfect balance between the maximum acceptable application performance that can be sacrificed for the purpose of extending the lifetime of the network. This operation balance point depends on the minimum requirements of the application in terms of performance and lifetime.

Despite the chosen point of balance, batteries constitute a limitation of the operational

lifetime of the system [91]. Advances in energy harvesting technologies have led to the possibility of realizing Energy Harvesting - Wireless Sensor Networks (EH-WSNs), making it possible to power wireless embedded devices by small-scale ambient energy [102]. Several sources of environmental energy can be harvested, such as solar power and wind power in outdoor deployments or heat from radiators and artificial light in indoor contexts. The key advantage of EH-WSNs with respect to battery-powered WSNs is that energy harvesting can continuously produce and provide the system with energy. As a result, the perpetual operation of the system is solely limited by hardware or software failures. Energy harvesting mitigates the need for battery replacements and, therefore, decreases the cost of maintenance that requires human intervention. Furthermore, energy harvesting constitutes an environmentally friendly energy source, as it uses renewable energy and reduces battery wastes.

1.2.1 Energy Sources

There are several sources of energy that have been considered for energy harvesting [16]. The sources can be classified in the following main categories: electromagnetic radiation, thermal energy and mechanical energy [46]. Table 1.1, taken from [98], shows the energy harvesting potential of several harvesting technologies.

Solar energy, out of the first category, is the most powerful source source for energy harvesting. The potential solar energy, available for harvesting, depends on various parameters, such as the geographical location of the node, the time of the day, the season of the year, the atmospheric conditions and the shadows created by the environment. A heavy cloud cover results in a drop in available energy of approximately an order of magnitude. When considering solar energy for supporting WSNs, it is important to consider that the energy is available for only one part of the day, while the sensor systems may be required to operate at the same level all the time during the day. Hence, the energy harvested during daytime should be stored for night time operation. Another potential energy harvesting source is artificial indoors light. Indoors light may be available during the night, depending on the nature of the indoors environment. However, a typical indoors light is orders of magnitude less powerful than direct sun light. Lastly, it should be noted that the efficiency of the energy conversion depends on the angle of the photo-voltaic panel to the light source. Other sources of radiation outside the visible part of the electromagnetic spectrum are typically unsuitable for energy harvesting, as they are very low-power and spread over the spectrum.

Thermal energy sources have also been considered for energy harvesting sensors. Radiators and pipes that carry hot water are straightforward options for thermal energy harvesting in indoors environments. Body heat is also considered as an option for energy harvesting in wearable sensors. The efficiency of conversion from a thermal source depends on the temperature difference between the sides of the thermoelectric

Table 1.1: Power density of harvesting technologies [98].

Harvesting Technologies	**Power Density**
Solar cells (outdoors at noon)	$15mW/cm^2$
Piezoelectric (shoe inserts)	$330\mu W/cm^3$
Vibration (small microwave oven)	$116\mu W/cm^3$
Thermoelectric ($10°$C gradient)	$40\mu W/cm^3$
Acoustic noise ($100dB$)	$960nW/cm^3$

transducer. In the cases of body heat and Low Surface Temperature (LST) radiators (the surface temperature of LST radiators is in the range $30 - 40°$C), the temperature difference and, therefore, the available energy for harvesting is very low. Significantly more power can be harvested by standard radiators that heat up to $50°$C.

The last category of energy sources, suitable for energy harvesting, is the group of mechanical energy sources. First, there are steady state mechanical sources that are constant over extended periods of time. These sources are based on air currents and water flows in either natural channels or inside pipes. Researchers have also investigated the possibility of using blood flow and breathing in humans as a source of energy for sensors that are related to the health sector. It is determined that significant power is available but the procedure is generally not acceptable by the subjects. Mechanical energy is also available from periodic motion. In this case, energy is only available for a short part of the cycle. For instance, vehicles or humans passing over piezoelectric energy harvesters can provide such energy. It has to be noted that energy harvesting from human motion, creates inconvenience to the humans. To avoid inconveniences, energy harvesting should be kept at low power levels. Another type of mechanical energy suitable for energy harvesting is vibration energy, which typically is available in indoor environments. The energy extracted from such sources depends on the frequency and the amplitude of the vibration. It also depends on the mass of the vibrating mass compared to the mass of the energy harvesting device, as the presence of the energy harvester affects the vibration. A last mechanical energy source is acoustic noise. However, there is far too little power available to extract, except for very rare cases of extremely high noise levels.

The vast majority of sources available for energy harvesting are characterized by spatial and temporal variations [58]. The electrical power generated by the transducer frequently changes over time in an unpredictable manner. To make matters worse, the energy harvested by different sensor nodes significantly variates even when they are placed in relatively close proximity. In practice, there are no guarantees that the energy will be available when needed. Energy storage constitutes a solution to this problem. Large capacitors are able to store energy that is sufficient for one or few measurements. Rechargeable batteries, such as Li-ion batteries, are energy buffers with substantially higher capacity. Hence, they are able to permit long-term (e.g. daily or weekly) energy

management.

1.2.2 Sustainable Operation

From a technical perspective, the system goal of EH-WSNs is fundamentally different from the one of battery-powered WSNs (i.e. to maximize the network lifetime). Indeed, as long as the harvested energy is more than or equal to the energy consumed, energy does not constitute a limitation on the lifespan of the embedded device. In this case, we say that a node operates at a *sustainable state* (Figure 1.4), also known as Energy Neutral Operation (ENO) state in literature [58]. Operating states where the harvested energy is much higher than the consumed energy are sustainable yet suboptimal, as the excess of energy is wasted instead of being used for increasing the performance of the system. Thus, any additional harvested energy should be used to improve the performance of the energy harvesting application. As a result, the system goal of EH-WSNs is twofold: sustainable operation constitutes the primary goal, while application performance represents the secondary goal whenever the energy input is sufficiently high to allow it. In other words, we aim at achieving the *maximal sustainable performance*: the desired operating state that the harvested energy is approximately equal to the consumed energy, since the system operates at a sustainable state while all the harvested energy is used to improve the system performance. Operating at this state, which in the literature is commonly referred to as *ENO-Max* [120], constitutes a foundational goal of WSNs that are powered by energy harvesting.

In practice, energy-efficiency remains a fundamental design goal of the system. Both goals of sustainability and application performance, require the system services and communication protocols to use the available energy in any efficient manner. All the considerations mentioned in the last paragraph of Section 1.1.1 remain perfectly valid. In addition to energy-efficiency, energy harvesting introduces the need for an additional design goal, namely *adaptability*. Due to the unpredictable and ever-changing nature of the energy harvesting sources, system services and communication protocols should be able to autonomously adapt their energy consumption to the available energy. Adaptation of the energy consumption of a service or protocol unavoidably leads to the adaptation of its performance.

Maintaining the energy consumption at the exact same level as the energy harvested, is generally impractical. A more practical implementation would use the energy buffer (e.g. a super-capacitor) to temporarily store unused harvested energy or to satisfy a sudden need for additional energy. Essentially, the system is required to maintain the residual energy of the energy buffer between a maximum and a minimum level, as shown in Figure 1.5. A decrease of the level of residual energy of the energy buffer, indicates that the consumed energy is more than the harvested energy. Similarly, an increase indicates the opposite. The energy consumption is adapted in accordance to

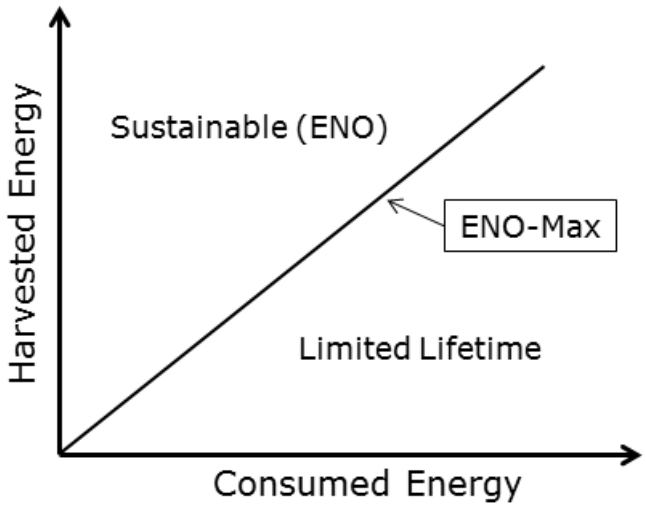


Figure 1.4: The operation of a sensor node is sustainable (ENO) if it harvests more energy than the energy it consumes. To maximize its performance (ENO-Max) the sensor node should put into use all the energy that it harvests.

the rate of change and the level of the thresholds.

An alternative practical implementation of sustainable operation with maximized performance follows a *harvest-before-consume* manner. In this approach the system periodically checks the residual energy in the buffer and performs a duty cycle only if the voltage is above a threshold, as shown in Figure 1.6. The threshold must account for the energy consumed in worst case scenario. In other words, it must guarantee that the energy consumed for the duty cycle will not leave the buffer with less than the minimum voltage required for the microcontroller to work.

1.3 System and Networking Issues

A sensor node constitutes a compact computing system that consists of several layers of abstraction. Their firmware includes drivers for the microcontroller, the radio and the peripheral hardware, as well as multiple protocols that loosely fall into the Open Systems Interconnection (OSI) layers of a communication system. In the firmware of a sensor node, one can find services that include communication protocols (e.g. routing / MAC protocols), end-to-end services (e.g. data encryption / node localization) and

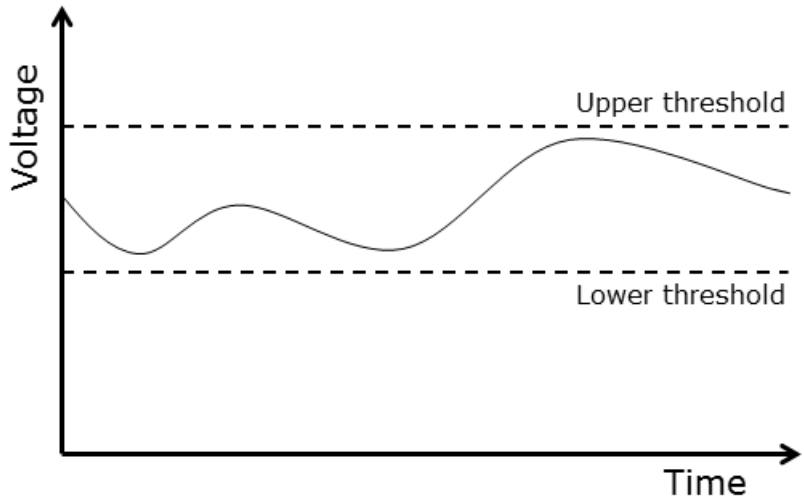


Figure 1.5: An energy buffer, such as a super-capacitor, can temporarily store unused harvested energy or to satisfy a sudden need for energy. Sustainable performance is achieved by maintaining the voltage of the energy buffer between two thresholds.

inter-node services (e.g. clock synchronization).

TinyOS [68] and Contiki [29] are compact operating systems that are designed for WSNs. They are build upon the principles of energy-efficiency, flexibility and innovation, in an attempt to meet the requirements of a wide range of low-power sensor applications. Specifically, they provide a set of various selectable services and protocols and due to their open-source nature, they are continuously extended with new features.

This section briefly reviews some key system issues that WSNs are challenged to face.

1.3.1 Node Localization

Many WSN applications and protocols depend on effective node localization [89]. Node localization is the problem of determining a node's position. A straightforward approach to this problem is adding a Global Positioning System (GPS) receiver to all the sensor nodes. Unfortunately, this solution may not always be feasible as it requires that the nodes have a clear line-of-sight to the GPS satellites, which is not always fea-

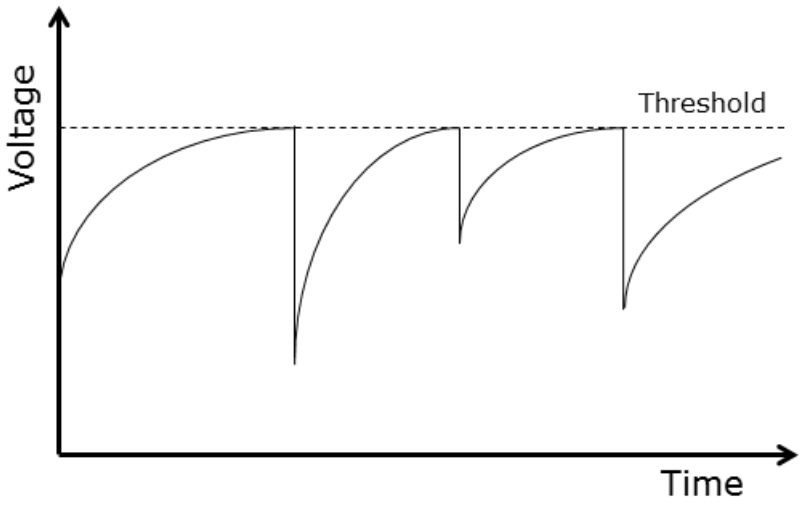


Figure 1.6: A sensor node can operate in a sustainable manner if it initiates a duty cycle only after it harvests enough energy to support it. The threshold voltage must be high enough to account for the energy consumption in the worst case scenario.

sible. In addition to that, GPS consumes energy and increases the cost and the size of the sensor. Alternative approaches include Received Signal Strength Indicator (RSSI) based methods, time based methods and techniques that use the angle of arrival. All these methods can estimate the relative position of a sensor from another sensor. Assuming that there are some nodes that already know their position (known as anchors or beacons), these methods are able to estimate exact positions.

The RSSI methods are based on theoretical and empirical propagation models. The idea is to measure the signal attenuation while knowing the transmission power and the antenna characteristics. Then, using models, it is possible to translate the signal loss into distance between the transmitter and the receiver. The second method is based on the difference between the time of transmission and the time of arrival. Since the signal propagation speed is known, the propagation delay can be directly translated into distance. Finally, the Angle-of-Arrival methods estimate the angle at which signals are received and use simple geometric relationships to calculate node positions. Combining multiple techniques can lead to more accurate estimations.

1.3.2 Clock Synchronization

Clock synchronization protocols are important to assure that different nodes in a distributed system have a common notion of time. Applications that need to correlate data over time require that the timestamps of each node's respective data have a common reference. In addition to applications, system services and protocols may also depend on clock synchronization. Clock synchronization in WSNs can be classified based on the following properties [108]. The communication model is either master-slave or peer-to-peer. In the master-slave model, all the slave nodes are synchronized to one master node. The peer-to-peer model is trading the simplicity and predictability for flexibility. Another important advantage of the peer-to-peer model is load balancing, as it leads to a more uniform energy distribution among the sensor nodes. Clock synchronization can be either internal or external. In external synchronization, all nodes are synchronized to a reference time. The reference time is typically the actual real-world time. In internal synchronization, the goal is to minimize the maximum difference between the readings of local clocks of the sensors. The synchronization can also be either probabilistic or deterministic. Deterministic synchronization algorithms guarantee an upper bound on the clock offset with certainty, whereas probabilistic synchronization algorithms provide a probabilistic guarantee on the offset with a failure probability that can be bounded or determined. The former approach requires more messaging and processing; hence, the probabilistic can better suit energy-constraint systems.

Data synchronization does not necessarily require clock synchronization. Apart from the approach of clock correction, the need for energy-efficiency has led to alternative approaches that leave the clocks untethered. In particular, the parameters that define the time offset of the local clock of a sensor to the clocks of each neighbor are saved in a reference table. Local timestamps are then compared and translated using these tables, essentially achieving a common notion of time between the nodes. The Receiver-to-Receiver synchronization model can be applied to generate such reference tables. This approach exploits the fact that if a message is broadcasted in the wireless medium, all its receivers will get it approximately the same time. Then, the receivers exchange the time at which they received the same message and compute their offset based on the difference in reception times.

1.3.3 System Security

The goal of security services in WSNs [26, 125] is to protect the sensed data and the resources from attacks and misbehaviors that jeopardize the intended operation of the sensor system. WSNs are vulnerable to various types of attacks in all the networking layers. These attacks can be classified into the following categories: attacks on secrecy and authentication, attacks on availability and attacks against service integrity. The

first category refers to the security requirement of confidentiality, which ensures that the data cannot be accessed by undesired nodes, authorization, which ensures that only authorized nodes are able to provide data to the system, and authentication, which ensures that the communication between two nodes is genuine. The second category is often referred to as Denial-of-Service (DoS) attacks and aims in keeping the WSN, partially or entirely, unavailable. The third category refers to attacks that aim to make the network accept false data values by compromised sensor nodes.

Research effort have been made in cryptography, key management, secure routing, secure data aggregation and intrusion detection, aiming to thwart these attacks. However, WSNs impose some unique challenges that need to be addressed. The selection of the appropriate cryptographic methods depends on the capabilities of the sensors' processor. Moreover, the design of security services must satisfy the resource constraints of the sensor nodes. Hence, there can be no unified solution for all WSNs.

1.3.4 Transmission Power Selection

Transmission power refers to the power level of the transmitted signal in a wireless communication. Wireless radios allow the selection between different levels of transmission power. In practice, the transmission range is strongly related to the transmission power. In simple words, a higher transmission power leads to a higher signal-to-noise ratio at the receiver and, therefore, to better chances for a successful packet reception. A larger transmission range implies more routing options and paths with fewer hops. On the other hand, a high transmission power is more energy consuming and translates into more contention for the wireless medium.

In [64] the authors present two local and distributed algorithms of selecting the transmission power in battery-powered WSNs, namely the Local Mean Algorithm (LMA) and the Local Mean of Neighbors Algorithm (LMN). In LMA, the nodes periodically adjust their transmission power so that the number of their neighbors converge to an adjustable attribute. LMN works similarly, but the transmission power adaptation aims to make the mean value of its neighbors' number of neighbors converge to an adjustable attribute. These algorithms are compared to global algorithms that make use of global knowledge and, hence, are able to achieve optimal solutions. The proposed algorithms cannot outperform the global ones, but they are practically implementable and scalable solutions.

Adaptive Transmission Power Control (ATPC) [74] is a protocol for transmission power adaptation for WSNs. It aims to minimize the transmission power levels while providing good link qualities and to dynamically change these power levels in order to address temporal fluctuations. The suggested algorithm has an initialization phase in which each pair of neighboring nodes communicate using different transmission rates

in order to build a prediction model that reflects the correlation of the transmission power and the link quality between them. During runtime, each transmitter selects the transmission power in accordance to the prediction model and the desired link quality. Then, the receiver provides feedback to the sender. Whenever the link quality is below the desired level or the link quality is good but the signal energy is so high that significant energy is wasted, the transmitter gradually adjusts the transmission power accordingly.

The impact of transmission power on EH-WSNs has been evaluated in [110]. The authors consider a multi-sink topology where the sensors communicate directly with one of the sinks (i.e. single-hop topology). Their work provide insight on how the transmission power affects several performance metrics such as network throughput density, data delivery ratio and throughput fairness. They conclude that these performance metrics can be maximized by appropriate transmission power adaptation.

1.3.5 Routing Protocols

The selection of the routing path in multi-hop wireless networks is not a trivial problem. The path that has the shortest distance between the sender and the receiver (i.e. minimum number of intermediate nodes) is often not the path that minimizes performance metrics, such as the end-to-end packet delay or the energy consumption of the network. Energy-aware protocols for battery-powered WSNs [3] aim to maximize the lifetime of the network by distributing the traffic among different paths. In EH-WSNs, routing protocols that are aware of the energy harvesting capabilities of the sensor nodes is a straightforward extension.

Voigt et al. conducted early routing investigations that consider sensor nodes that are powered by alternative sources [123] [122]. First, they presented a solar-aware routing protocol that preferably routes traffic via nodes that are powered by solar energy harvesting [123]. The protocol identifies and establishes the shortest path between the source and the sink. Generally, all data packets propagate over this path. However, the source and maximum one of the intermediate nodes may choose to forward the data to a node that is solar-powered rather than a node on the shortest path. This way the protocol avoids loops. Simulations verify that solar-aware routing provides significant energy savings in many scenarios. Then they considered cluster-based WSNs, where the cluster heads are responsible for performing energy-intensive tasks, including routing traffic to the sink. The authors extend Low Energy Adaptive Clustering Hierarchy (LEACH) [49], the well-known cluster-based protocol, to become solar aware. The proposed extension, named Solar Low Energy Adaptive Clustering Hierarchy (sLEACH) [122], integrates a simple yet effective idea. Regardless of the method used to decide the cluster heads, the solar-driven nodes that have a high remaining energy level shall have higher probabilities of becoming a cluster head. The paper shows

that integrating solar awareness into LEACH increases the lifetime of a sensor network significantly.

The aforementioned works provide initial insight on the benefits of designing protocols that are aware of the energy-harvesting capabilities of the sensor nodes. However, energy-harvesting is treated as a binary feature. Sensor nodes either have energy-harvesting capabilities or not. Hence, all nodes with energy-harvesting capabilities are treated equally. More recent works in EH-WSNs consider the spatial and temporal variations of the availability of ambient energy.

Reference [72] addresses the problem of choosing the most energy-efficient route in EH-WSNs. In particular, they route each packet over the path that minimizes a cost metric that depends on the nodal replenishment rate, the residual energy on the rechargeable battery and the energy requirements for the transmission and reception of the packet. All three parameters are shown to be essential for an energy-efficient routing metric. The proposed routing metric can be incorporated into existing routing schemes (e.g. proactive or on-demand methodologies).

Lattanzi et al. verified the importance of taking into account the energy profile of each individual sensor node when deciding the routing path. In their work [66], they evaluate four different routing algorithms which gradually integrate awareness of an additional energy-related factor. The first algorithm, named *Minimum Path (MP)*, routes traffic on the path that minimizes the number of hops (i.e. number of intermediate relay nodes). The second algorithm, named *Randomized Weighted Minimum Path (R-WMP)*, takes into account both the number of hops and the power requirements of each link that varies due to the distance between each transmitter and receiver. The third algorithm, named *Randomized Minimum Path Energy (R-MPE)*, routes the traffic over the path that minimizes the energy consumption to reach the sink. The forth algorithm, named *Randomized Minimum Path Recovery Time (R-MPRT)*, routes the traffic over the path that minimizes the energy recovery time. The energy recovery time is defined as the time required for a sensor node to harvest energy in order to recover from relaying the respective packet. Their results show that each additional energy information taken into account gradually increases the performance of the network.

Distributed Energy Harvesting Aware Routing (DEHAR) [56] is another routing protocol that routes traffic based on the energy profile of each node. DEHAR calculates the shortest paths and then applies penalties to each path based on the residual battery level and the energy harvesting rate of each intermediate node. The best path is selected based on a metric that they call *energy distance*. Additional penalties are added in an attempt to avoid routing dead ends. DEHAR indirectly takes into account the energy consumption requirements of each path by assuming that the shortest path to the sink is also the least energy consuming path. This assumption is true only when all sensors are using the same transmission power. The proposed routing algorithm is shown to find sustainable paths from any source to the sink.

Table 1.2: Factors integrated into routing metrics for EH-WSNs.

	HC	RE	TxP	HR	CE	DC
Solar-Aware [123]	✔	-	-	partially	-	-
sLEACH [122]	✔	-	-	partially	-	-
Energy-Aware [72]	✔	✔	✔	✔	-	-
MP [66]	✔	-	-	-	-	-
R-WMP [66]	✔	-	✔	-	-	-
R-MPE [66]	✔	-	✔	-	-	-
R-MPRT [66]	✔	✔	✔	✔	-	-
DEHAR [56]	✔	✔	-	✔	-	-
GREES [135]	✔	✔	✔	✔	✔	-
Opportunistic [35]	-	✔	-	✔	-	✔

Wireless communications are characterized by lossy links. The Bit Error Rate (BER) is significantly larger than the case of wired links due to the relatively high levels of noise and interference in the wireless medium. Considering the spatial and temporal variation of channel errors, a routing protocol can maximize the probability of a successful transmission by forwarding traffic over paths that are less lossy. This was investigated by the authors of [135]. They presented a routing protocol named *Geographic Routing with Environmental Energy Supply (GREES)*. GREES evaluates each link based on a metric that is a function of the distance to the final destination (equivalent to hop count), the residual energy, the energy harvesting rate, the energy consuming rate and the wireless link quality. Simulations on an environment that models channel errors verify the importance of considering this factor for the selection of the path.

Reference [35] incorporates an opportunistic scheme. Instead of evaluating a path based on link parameters, the next forwarder is decided opportunistically. The transmitter broadcasts the frame and any sensor that is nearer to the sink than the sender and happens to be available for forwarding traffic, rebroadcasts the frame. This approach exploits the sleeping schedules of the sensor nodes, also known as Duty Cycle (DC), which will be presented further in the following section.

Table 1.2 summarizes the factors considered by routing protocols designed for EH-WSNs. These factors include: the hop count (HC), the residual energy (RE), the transmission power (TxP), the harvesting rate (HR), the channel errors (CE) and the DC.

1.4 Duty Cycles and Medium Access Control

The Medium Access Control (MAC) layer plays a key role in wireless sensor networks. It is primarily responsible for the establishment of communication links between nodes, that are vital to form the network infrastructure. The MAC scheme then regulates the access to the shared wireless channel by multiple nodes. In addition to that, the MAC protocol plays a key role in the design of energy-efficient WSNs. Since the radio of a sensor node consumes the highest amount of power [7], the main method of preserving power is to duty cycle the node. Duty Cycles (DCs) are materialized by alternating the node between active and sleeping states, where the node is operational in the active state and shut down in the sleeping state.

For a communication link to be established, both the receiver and the sender need to be simultaneously in an active state. Here, an important distinction needs to be made. The sink node has no energy constraints and, therefore, there is no need to duty-cycle its radio. As a result, unless there is a need for transmitting, the radio is always on receiving mode, similarly to traditional wireless networks, e.g. Institute of Electrical and Electronic Engineers (IEEE) 802.11 Distributed Coordination Function (DCF) [55]. Therefore, in the case of single-hop star topologies, establishing the link does not constitute a particular challenge. A duty-cycling sender will always find the receiver available to receive traffic. In multi-hop topologies, on the other hand, both the sender and the receiver are duty cycling. This poses a particular problem of finding a rendezvous point between a sender and receiver, in which both of the nodes are in an active state and a communication link can be established.

1.4.1 Duty-Cycling Sender – Always-On Receiver

In this section, we consider a link where the sender is duty-cycling its radio, while the receiver is in listening mode, by default. Thus, upon a wake-up event, the sender and the receiver can directly communicate. For the time the sender is on active mode, the link resembles traditional wireless networks and ALOHA-inspired [2] Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) wireless communication protocols are applicable (e.g. the beacon-free version of IEEE 802.15.4 [54]). IEEE 802.11 DCF [55] is also applicable (see the low-power Wi-Fi of RTX4100 [100]).

In the context of EH-WSNs, [34] considered and compared four fundamental MAC schemes, namely Slotted Carrier Sense Multiple Access (CSMA), Unslotted CSMA, ID Polling and Probabilistic Polling. Slotted CSMA divides the time into timeslots and the sensors contend for each timeslot. Each node first senses the medium and if it is free then it transmits the frame. Then, the node goes into a charging state until the next packet generation. The Unslotted CSMA works similarly to a standard

CSMA/CA protocol implementing backoff mechanisms to avoid collisions. Again, the sensor goes into a charging state whenever there is no packet for transmission. Next is the ID Polling scheme where the sink randomly requests for a packet from a sensor using explicitly its unique ID. In this scheme a sensor is in a charging state whenever there are no data for transmission and in a listening state in case there are. Lastly, in Probabilistic Polling the sink defines and broadcasts the probability that a sensor would transmit data. Upon receiving a polling packet each sensor generates a random number and transmits accordingly. The authors of the paper conclude that Unslotted CSMA and Probabilistic Polling perform best. In fact, Unslotted CSMA performs better for a low number of sensor nodes and when this number exceeds some threshold Probabilistic Polling dominates.

The simplicity of the scenario studied in this section, allows the abstraction of the MAC layer from the duty-cycle selection algorithm. In practice, the MAC layer is abstracted to a simple number that indicates the percentage of time a sensor spends in active mode. For example, a 10% duty-cycle indicates that the node spends one every ten time slots in active mode and the remaining nine time slots in sleeping mode. In this context, a line of works attempts to optimize the duty-cycle without being concerned with the complexity of the MAC layer.

The duty cycling algorithms suggested in [58] consists of three parts. The first part tracks past energy input profiles and uses them to identify patterns and predict future energy availability. The second part computes the optimal duty cycle based on the energy prediction. The third part handles the expected prediction errors by dynamically adapting the duty cycle of the sensor in response to the observed energy harvesting in real time. The final part operates as follows. The harvested energy is measured over fixed timeslots and the excess of energy in each slot is calculated. Whenever the excess of energy is negative, the duty cycle is decreased for the future slots giving the battery the opportunity to charge. If the excess of energy is positive, the duty cycle is increased in order to utilize that energy to increase the performance.

The approach of [120] does not use energy prediction profiles. Instead, the authors dynamically adjust the duty cycle aiming to maintain the ENO-Max condition. The dynamic duty cycle adaptation algorithm is based on the optimal tracking problem, addressed by adaptive control theory. It refers to the problem of applying external control to a dynamical system in order to keep some output variable at a desired value. The authors choose to map the duty cycling problem to the version of the optimal tracking problem named linear-quadratic. In this version, it is assumed that the dynamics of the system (i.e. battery level, harvested energy, power consumption) are linear while the cost function to be minimized (i.e. average difference of the current and the initial battery level) is quadratic.

1.4.2 Duty-Cycling Sender – Duty-Cycling Receiver

In this section, we consider a link where both the sender and a receiver are duty-cycling their radio. This introduces the challenge of finding a moment in time that both the sender and the receiver are active and a communication link can be established. MAC schemes for WSNs take a synchronous or asynchronous approach to solve this problem. Figure 1.7, depicts the synchronous and asynchronous paradigms for coordinating the receiver and the transmitter in duty-cycled wireless communications.

In protocols that follow the *synchronous* approach, like Sensor-MAC (S-MAC) [132], T-MAC [21] and DSMAC [73], nodes organize the active and sleeping states to align. Synchronous schemes can be based either on contention or on reserved timeslots. In both cases, a portion of the active state is used to synchronize all the nodes to a global active/sleep schedule. Synchronous schemes are quite tolerant to schedule misalignment, however, they still require a globally synchronized schedule, which creates an additional energy overhead. Additionally, synchronous protocols have a cost associated with the creation and maintenance of the schedule. Furthermore, the coupling of nodes via a global clock also hinders a node's ability to have a fully independent duty cycle, so that each node can adapt, in a fully distributed way, to the current surrounding conditions.

S-MAC [132] was a milestone protocol for the synchronous class. S-MAC defines a MAC protocol in which neighboring nodes form virtual clusters that share a common sleeping schedule. The time is divided in active and sleeping periods. All the sensor nodes of the cluster communicate in the active period, essentially saving energy during the sleeping period. The activity periods are scheduled by periodical Synchronization (SYNC) packets between the neighbors.

The IEEE 802.15.4 [54] standard defines the Physical (PHY) and MAC layers for Low Rate - Wireless Personal Area Networks (LR-WPANs). The beacon-based version of the MAC protocol, incorporated inside the IEEE 802.15.4 standard, follows the synchronization paradigm. In particular, the standard defines two types of nodes, the Reduced Function Devices (RFDs) and the Full Function Devices (FFDs). RFDs can only act as end-nodes. FFDs, on the other hand, have full MAC functions and are able to act both as end-nodes and as network coordinators. The communication between the nodes is achieved as follows. The coordinator periodically sends one beacon, which defines a superframe and is used for synchronization. The superframe which consists of three portions and the beacon includes information about their duration. There are two active portions that are divided into fixed slots. The first active portion is the Contention Access Period (CAP), where nodes contend for channel access based on a slotted CSMA/CA scheme. The second active portion is a Contention Free Period (CFP), where nodes transmit without contending for channel access in Guaranteed Time Slot (GTS) assigned by the coordinator. Then, there is an inactive portion, that is used by the

coordinator to sleep and save energy.

Asynchronous schemes do not require synchronization, as the nodes sleep and wake up independently of the others. This leads to the need of techniques on deciding a rendezvous point for nodes to communicate. There are two fundamental asynchronous techniques, namely the sender- and the receiver- initiated.

The basic technique used in a sender-initiated asynchronous MAC scheme is called preamble sampling, where the sender transmits a preamble to indicate that there is a pending need for communication. The receiver wakes up occasionally into the active state, to listen to such a preamble transmission. Once the preamble is detected, the receiver replies with a positive acknowledgment to the sender when the preamble transmission stops. This establishes a communication link between the sender and receiver. Most notable examples of MAC protocols that are based on the sender-initiated paradigm are WiseMAC [32], Berkley MAC (B-MAC) [95] and Short Preamble MAC (X-MAC) [13].

B-MAC constitutes a milestone complete implementation of the sender-initiated approach [95]. X-MAC uses a short strobed preamble to further improve upon the weaknesses of B-MAC [13]. Instead of a long preamble, X-MAC is transmitting multiple short preambles that contain addressing information. The appropriate receiver is given with enough time to interrupt the series of short preambles with a special packet named *pre-ack* that indicates that it is ready to receive the data. A variant of X-MAC is implemented in the TinyOS embedded operating system [68]. Currently, X-MAC is the most widely used sender-initiated scheme. A thorough survey of sender-initiated schemes is performed in [14], concluding with a guideline to select MAC schemes for a given application.

In contrast to the preamble sampling technique in sender-initiated schemes, receiver-initiated schemes use another approach to asynchronous communication: instead of long preambles, the sender listens to the channel, waiting for small beacons transmitted by the receiver. The receiver transmits the beacons in a period that is defined by its duty cycle, and is used by the sender to synchronize with the receiver. The receiver-initiated paradigm was originally introduced by Receiver Initiated Cycled Receiver (RICER) [71] and made popular by Receiver Initiated MAC (RI-MAC) [107]. In Chapter 2, we present an in-depth survey of the receiver-initiated paradigm of communication.

The complexity of the communication between duty-cycling nodes, especially in the asynchronous schemes, does not allow the abstraction of MAC layer during duty cycle optimization. Duty cycle optimization is highly correlated with the optimization of the MAC layer parameters and, therefore, it highly depends on its mechanics.

The authors of [87] analyze and evaluate several sleep and wake-up strategies. In par-

ticular, their study includes a battery-state-based sleep and wake-up strategy where the sensor node decides to switch between sleeping and active mode based on the normalized battery capacity. It also includes a queue-based strategy where switching between sleeping and active mode is based on the number of queued packets. Similarly, they consider the channel state-based strategy where the decision is determined by the status of the channel and the solar-radiation-based strategy where the decision is made based on the energy harvesting rate. In addition to those fundamental strategies, the authors also consider hybrid strategies that combine two or more of the fundamental strategies. Their conclusions suggest that there is not optimal strategy; instead, there are trade-offs and some strategies can be better than others on different performance metrics. Despite the fact that they assume multi-hop networks with links of duty-cycling senders and receivers, they do not consider a MAC protocol that guarantees coordination between transmitters and receivers by following one of the three communication paradigms. Instead, a transmitter may send a frame to a sleeping node.

1.5 Scope and Contributions of the Dissertation

The primary focus of this dissertation lies on the MAC layer of EH-WSNs. In short, our goal is to improve the performance of the MAC layer towards the principles of sustainability and application performance. As mentioned previously in this chapter, a sensor node can have a long-term sustainable operation only if it is able to adapt the energy it consumes to the unpredictable and ever-changing ambient energy that it can harvest. High application performance, on another hand, requires efficient use of the energy resources in a twofold sense. Firstly, every feature and protocol running in the sensor node should consume the least amount of energy possible, not to compromise the energy-efficiency of the system. Secondly, all harvested energy should be used and not wasted in full energy buffers. Therefore, the goals of a MAC protocol can be summarized as energy-efficiency and adaptability.

The scope of the dissertation is defined by the assumed network and application characteristics, as well as the assumed properties of the environment the network is deployed. Unless otherwise noted, the assumptions of this work are the following.

- We assume sensor nodes that are powered by ambient energy, i.e. EH-WSNs. Nevertheless, energy-efficiency is a goal of a WSN regardless the energy source (batteries or energy harvesting). Therefore, the usefulness and application of some of the proposed MAC features extend beyond EH-WSNs.
- We assume that the ambient energy is unpredictable with spatial and temporal variations. The assumption of unpredictability is valid in the vast majority of the potential energy sources. Sometimes, high level predictions are possible (e.g.

there will be more solar energy during the summer than the winter). However, the exact amount of available energy cannot be predicted accurately.

- We assume that the ambient energy is uncontrollable. Natural energy sources are generally uncontrollable. Additionally, artificial energy sources are also uncontrollable from the perspective of a WSN. For example, consider a sensor node powered by artificial indoors light. While the energy source is controlled by humans, their behavior cannot be controlled by the WSN.

- We assume continuous periodic traffic generation. In addition to monitoring applications, this assumption covers event-tracking applications that generate periodic traffic to report negative acknowledgments.

- We assume that there are no mobile nodes in the network. All sink and sensor nodes are considered static.

- We assume links where both the sender and receiver are duty-cycling to save energy. Links with the always-on receivers are briefly considered in Chapter 9.

1.5.1 Key Contributions

Given the above assumptions, our research on the MAC layer of EH-WSNs resulted in the development of an experimental receiver-initiated MAC protocol that we named On Demand MAC (ODMAC). The purpose of ODMAC is not to provide full MAC layer functionality covering all the elements a MAC protocol is meant to address. Instead, the goal of ODMAC is to be a testing platform that would allow the experimentation and evaluation of different optimization features. Indeed, the key contributions of ODMAC lie in its unique optimization features. The key features of ODMAC are (i) adaptive duty cycles, (ii) opportunistic forwarding, (iii) collision avoidance and traffic differentiation with Altruistic Backoff (AB) and (iv) the Receiver Authentication Protocol (RAP). The performance and behavior of ODMAC and its features are analyzed and evaluated using mathematical models, simulations in MATLAB [79] and OPNET [88] and experiments in real testbeds (based on the eZ430-rf2500 [115] wireless development platform). Furthermore, ODMAC is analytically compared to two state-of-the-art MAC protocols that are widely used in academia and in a large-scale industrial network, respectively.

Further contributions of this dissertation include a survey of all the MAC protocols that follow the receiver-initiated paradigm of communication and the development of an energy harvesting Carbon Dioxide (CO2) sensor node that is based on IEEE 802.11 [55], more commonly known as Wi-Fi.

1.5.2 Structure of the Dissertation

The remainder of the dissertation is structured as follows. Chapter 2 focuses on the receiver-initiated paradigm of communication and provides an in-depth survey of all the MAC protocols that are built upon it. The survey aims to provide the reader with an overview of the features that various protocols, including ODMAC, implement and concludes with a discussion on the conditions that some particular features are more suitable than others.

The following chapters focus on ODMAC. Chapter 3 introduces the protocol and its features, while Chapters 4 to 8 evaluate its performance. In particular, Chapter 4 analyses and evaluates the adaptive duty cycles and opportunistic forwarding. The evaluation is twofold. The first part is based on an analytical model and the second part is based on simulations in OPNET. Chapter 5 evaluates the performance of collision avoidance with AB. A comparison of AB with the state-of-the-art collision avoidance mechanism demonstrates its energy-efficient character. The ability of AB to prioritize important data packets via traffic differentiation is also evaluated. Chapter 6 introduces the beacon replay attack and evaluates RAP. The effectiveness of the protocol to counter the beacon replay attack is verified formally using two protocol verification tools. Additionally, the chapter demonstrates the trade-off between the energy consumption overhead of the scheme and the level of security it provides.

Chapter 7 analytically compares ODMAC to two state-of-the-art MAC protocols, namely X-MAC [13] and Inter-Meter Reading + (IMR+). The former is a sender-initiated protocol that is widely used in academia, as it is implemented in TinyOS [68]. The comparison of the two protocols evaluates the suitability of the two asynchronous paradigms in an energy harvesting context. The latter is an industrial protocol that is currently used in a large-scale commercial WSN. The comparison focuses on the structure of the existing network, considering the potential upgrade of the network with energy harvesting sensor nodes.

Chapter 8 presents the implementation of ODMAC on the eZ430-rf2500 [115] wireless sensor nodes. Using the presented implementation, ODMAC is evaluated in a real testbed. The experiments demonstrate sustainable operation for different levels of power input and evaluate AB in scenarios that multiple nodes contend for the wireless medium and in scenarios with traffic of different urgency.

Chapter 9 moves the attention to links where only the sender is duty-cycling while the receiver is always active. In this context, we first present the development of energy harvesting CO2 sensor node that is based on Wi-Fi. Then, we discuss an ambitious alternative way to transmit traffic in an energy-efficient manner, by exploiting timing channels.

Lastly, Chapter 10 concludes the dissertation and discusses issues that are open for future research.

1.5.3 Publications

The research that is presented in this dissertation resulted in several publications [25, 36–42, 121] or manuscripts that are submitted for publication [43].

Specifically, Chapter 2 and Section 1.4.2 contain material submitted for publication in [43]. Sections 4.2 to 4.4 are based on content published in [39]. Section 3.2, Section 3.3 and Section 4.6 are based on content published in [38]. Section 3.4, Chapter 5 and Section 8.3.5 contain material submitted for publication in [41]. Section 3.5 and Chapter 6 contain material published in [25]. Section 7.2 contains material published in [40]. Section 7.3 contains material published in [121]. Chapter 8 is based on content published in [36]. Section 8.3 contains experiments that were also demonstrated in [37]. Lastly, Section 9.2 contains material published in [42].

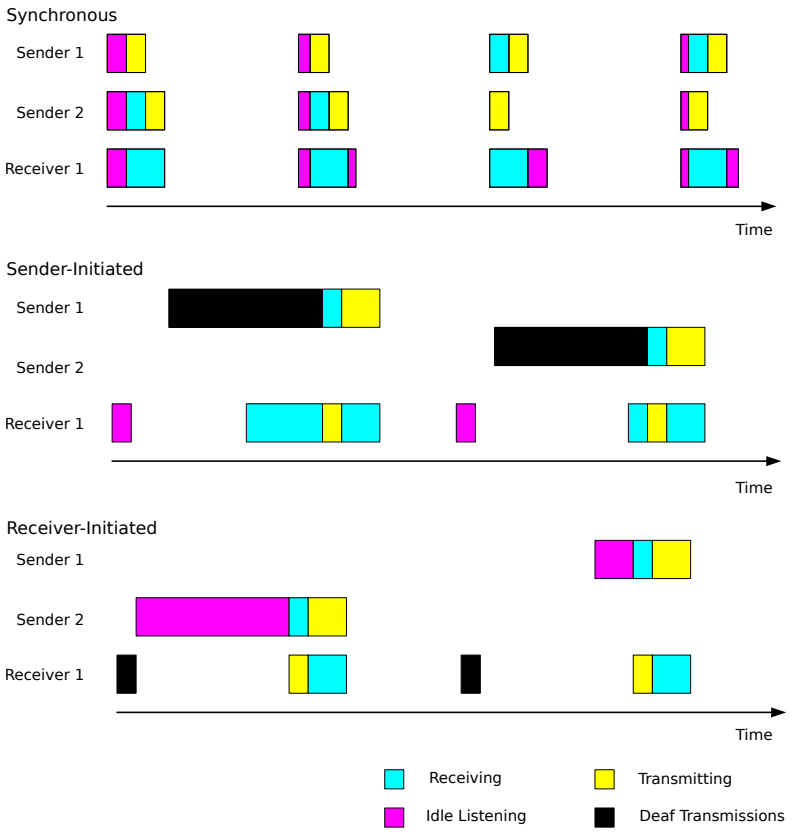


Figure 1.7: The three paradigms of communication between duty-cycling nodes, from top to bottom: synchronous, sender-initiated and receiver-initiated. *Idle Listening* and *Deaf Transmissions* indicate sources of energy consumption where the node is active, receiving or transmitting data respectively, while the other side of the link is in sleeping mode.

CHAPTER 2

A Survey on Receiver-Initiated MAC Protocols

2.1 Introduction to the Survey

In energy-efficient WSNs with links of duty-cycling senders and receivers, the MAC protocol faces the problem of finding a moment in time that both the sender and the receiver are in an active state, so that a communication link can be established. As introduced in Section 1.4.2, MAC schemes for WSNs can be classified into three basic paradigms of communication between duty-cycling nodes. In protocols that follow the synchronous paradigm, nodes organize the active and sleeping states to align. In protocols that follow the the sender-initiated asynchronous paradigm, the sender transmits a preamble to indicate that there is a pending need for communication. The receiver wakes up occasionally into the active state and if a preamble is detected, it remains in active mode and the communication link is established.

This chapter surveys the receiver-initiated asynchronous paradigm. In protocools that follow the receiver-initiated paradigm, the communication is initiated by the receivers who periodically transmit beacons that state their availability to receiver data. Contrary to the sender-initiated approach, the sender silently listens to the channel, waiting for

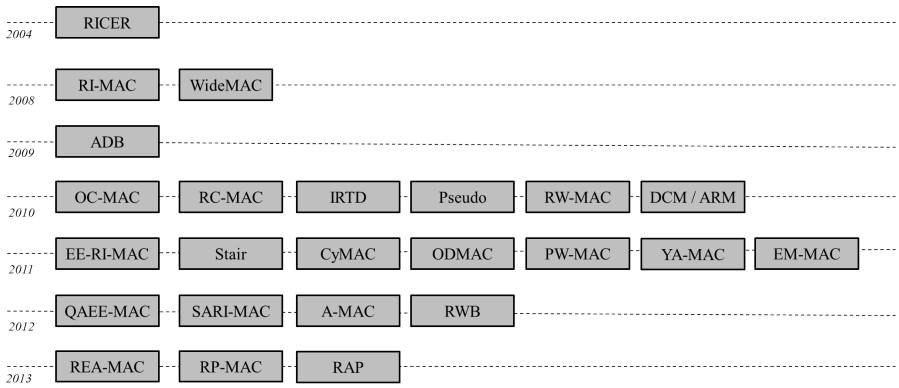


Figure 2.1: Chronology of Receiver Initiated MAC protocols.

the reception of a beacon. The receiver-initiated paradigm was originally introduced by Lin *et al.* in 2004 (RICER [71]) and made popular by RI-MAC [107] in 2008. Since the publication of RI-MAC, several MAC protocols that build on the receiver-initiated paradigm have been proposed (see Figure 2.1).

The survey first highlights the key design challenges a receiver-initiated MAC protocol should address. Then, keeping in mind these challenges, we survey all the MAC protocols that fall under the receiver-initiated category, analyzing and organizing them according to common features and design goals. ODMAC and RAP are also included in the survey. In Section 2.2, we present the challenges that receiver-initiated MAC protocols are meant to deal with. Section 2.3 classifies and presents all the existing MAC protocols that are based on the receiver-initiated paradigm. The classification is based on the most prominent or novel features that each protocol implements. In Section 2.4, we discuss, summarize and compare the surveyed protocols, focusing on how appropriate they are for specific application classes. Lastly, Section 2.5 concludes the survey.

2.2 Challenges for Receiver-Initiated MAC Protocols

MAC protocols are typically responsible for controlling the communication between two nodes over a link and for coordinating multiple nodes that share the same medium. Some of these tasks carry over from regular wireless networks, for example *protocol overhead* has to be taken into account: both activating the radio transceiver and producing unnecessary data exchange would lead to performance degradation, therefore the size and number of packets sent should be kept to a minimum. Naturally,

the goals (and hence the definition of performance) are different between regular and sensor-based wireless networks; most likely in the first case the dominating factor is throughput while in the second is energy preservation and network lifespan, but in the end the same concept still applies. *Channel error handling* is also a well-known problem with fairly standard solutions. Acknowledgments, re-transmissions, Cyclic Redundancy Checks (CRCs) (or authentication code if security is involved) are pretty much standard and consolidated techniques used in many MAC protocol for wireless networks.

In addition to these, new challenges are introduced. For example, receiver-initiated MAC protocols for WSNs have to deal with the fact that wireless sensor nodes are duty cycling between active and sleeping states to save energy. This produces new challenges for the MAC layer, such as minimizing the energy overhead for synchronizing the transmitter and the receiver. Moreover, broadcasting becomes less trivial, as some of the nodes could be sleeping at any given time.

In this section, we summarize the important challenges of the MAC layer for duty-cycling nodes that are following the asynchronous receiver-initiated paradigm.

2.2.1 Idle Listening

According to the receiver-initiated paradigm, each node with data to transmit enters an active state and listens to the medium for a beacon from the intended receiver. Until the time when the receiver wakes up from its sleeping state and transmits the beacon, the transmitter is essentially wasting energy listening to the channel without receiving any useful data. At the receiver's side, after every unanswered beacon, the node also wastes energy listening for a reply. This energy overhead is named *idle listening* and constitutes a weakness that is associated particularly with the receiver-initiated paradigm. As a result, there is significant literature work, focused on mechanisms to mitigate it.

2.2.2 Collision Avoidance

Contention-based MAC protocols for wireless communication are known to be vulnerable to colliding transmissions, as a radio that is transmitting is unable to detect other transmissions in the wireless medium. Collisions decrease the systems performance and are also a source of energy wastage. Protocols following the asynchronous receiver-initiated paradigm, may be either vulnerable or resilient to collisions depending on the topological structure of the network and the duty cycles of the nodes. This phenomenon rises because of the fact that beacons constitute indirect transmission

timeslots. When the beacon transmission rate is significantly higher than the data transmission rate, the stochastic selection of a beacon acts as an indirect proactive collision avoidance mechanism (random channel access). Yet, there is always the chance for multiple nodes to select the same beacon / timeslot. Hence, when the beacon and data transmission rate is at a similar order of magnitude, collisions are significantly increased and the system is lead to a state where the receivers are flooded with more transmissions than they can handle. This scenario appears either in topologies when few receivers have to handle large numbers of transmitters or in the case of low duty cycle receivers serving high duty cycle senders. The latter case requires active Collision Avoidance (CA).

2.2.3 Adaptive Duty Cycling

The dynamic adaptation of the duty cycles can significantly improve the energy efficiency of the system. A MAC protocol with adaptive duty cycles, that is aware of the structure of the topology, the traffic conditions or the resources of the nodes, can more efficiently use the available energy. For example, the nodes that are closer to the sink typically have more forwarding tasks rather than the nodes that further away. Additionally, independent duty cycle adaptation is vital for WSNs that are powered by harvested ambient energy. As introduced in Section 1.2.2, the system goal of such networks is to operate at a state where the consumed energy is on average equal to the harvested energy. Due to the chaotic nature of the environmental energy sources, the duty cycles of the node need to be frequently and independently adapted.

2.2.4 Quality of Service

Different types of packets can coexist within the network. According to the requirements of the overlying application, or even the protocol itself, each class of frame might require different handling. For example high priority messages might be relayed before low priority ones, frames could be reordered to meet delay bounds or again control messages could take precedence over data messages to ensure the correct functioning of the network. All these kind techniques fall under the generic definition of Quality of Service (QoS).

2.2.5 Broadcast Communication

Although trivial for typical MAC protocols for wireless communications, broadcast communication constitutes a challenge in networks of nodes that are duty cycling in an

asynchronous manner. Since the sleeping and activity periods of nodes is not synchronized in time, it is unlikely for a transmitter to find a moment where all the nodes are awake and ready to receive a broadcast transmission. Assuming a system-wide known maximum beacon period, this issue can be solved by replacing a broadcast communication with multiple unicast transmissions. Nevertheless, there is work in literature on other efficient ways to overcome this challenge.

2.2.6 Security

Sensor networks are vulnerable to attacks which are associated with the wireless medium. Wireless channels can be easily eavesdropped and traffic can be easily injected or altered. Attackers are not limited by the resource constraints of sensor nodes and can interact with the network from afar, using much more powerful equipment. Moreover, sensor networks may be deployed in psychically insecure environments and sensor nodes are vulnerable to resource depletion attacks and tampering in general. The security of the MAC layer is fundamental for the security of the system.

2.3 Receiver-Initiated MAC Protocols

The receiver-initiated paradigm of asynchronous communication for duty cycling nodes was introduced by RICER [71] in 2004. In 2008, Koala [86] defined a receiver-initiated mechanism, named Low Power Probing (LPP), which uses the receiver-initiated paradigm for the purpose of waking up the sensor nodes, while it is not involved in the actual data transfer. Later, the receiver-initiated paradigm was popularized by RI-MAC [107], which triggered vast research that builds upon the paradigm and optimizes its performance.

Each protocol that extends the receiver-initiated paradigm focuses on one or more of the challenges enumerated in Section 2.2. The rest of the section and the surveyed protocols are organized as follows. First, we present the receiver-initiated paradigm, as it was introduced by RICER [71] (Section 2.3.1). Section 2.3.2 surveys the receiver-initiated MAC protocols that provide an extension of the paradigm with focus on the fundamental challenges of *Idle Listening* and *CA*. The focus in Section 2.3.3 is on mitigating *Idle Listening* in the particular direction of predicting the following wake-up of the receiver. Section 2.3.7 surveys protocols that focus on the direction of using multiple channels to distribute the transmissions and decrease the contention. The remaining subsections can be directly mapped to a respective challenge in focus, as listed in Section 2.2. Table 2.1 summarizes the organization of the protocols according to their key design feature.

Table 2.1: A list of the surveyed protocols organized by their prevalent feature.

Feature	Protocols
Receiver-initiated	RICER
Basic extensions	RI-MAC, OC-MAC, RC-MAC, IRDT, EE-RI-MAC, A-MAC, REA-MAC, RP-MAC
Wake-up prediction	WideMAC, Pseudo, RW-MAC, PW-MAC
Adaptive duty cycling	Stair, ODMAC, SARI-MAC
Quality of service	CyMac, QAEE-MAC
Broadcast support	ADB, YA-MAC, RWB
Multi-channel extensions	DCM, EM-MAC
Security	RAP

2.3.1 The Receiver-Initiated Paradigm of Communication

The receiver-initiated paradigm operates as follows. Each node periodically wakes up to check for incoming data. After each wake-up event, a *beacon* is broadcasted. This beacon announces to the neighbors that it is ready to accept incoming data. After the beacon has been transmitted, the receiver continues to listen to the channel for a short period of time. Whenever a node with data ready to be sent enters the active state, it listens silently to a beacon from the intended receiver. Once the beacon is received, the sender immediately starts transmitting the data, and waits for a time period to receive a frame which acknowledges the reception of the data. If there is no incoming data from the sender after transmitting the beacon, the receiver enters the sleeping state. Both the sender and receiver, then resume their cycles.

In comparison to the sender-initiated paradigm, the receiver-initiated communication paradigm significantly reduces the amount of time for which a pair of nodes occupy the channel, allowing more contending nodes to communicate with each other, increasing the capacity and throughput of the network. It is more efficient in detecting collisions and recovering lost data, because access to the channel is mainly controlled by the receiver. Since receivers only wait a short period of time for incoming data, after beacon transmission, overhearing is greatly reduced [40, 71, 107].

2.3.1.1 Receiver Initiated CyclEd Receiver (RICER) [71]

Beyond introducing the paradigm, RICER also defines several features that improve the performance of the protocol. First, it uses a random delay between the reception of the wake-up beacon and the data transmission to avoid collisions. Furthermore, the authors note that a significant reduction of the energy consumption can be achieved by intro-

ducing multiple potential receivers. However, no particular receiver selection policy is specified, as it is considered a task of the routing layer. Lastly, a semi-synchronous mode is defined to decrease the energy consumption. With globally known duty cycles, nodes can keep record of the wake-up times of neighboring nodes to predict with approximation the upcoming wake-up.

2.3.2 Basic Extensions

RI-MAC [107] and other MAC protocols build upon the paradigm with features that optimize their performance.

2.3.2.1 Receiver-Initiated MAC (RI-MAC) [107]

RI-MAC builds on the receiver-initiated paradigm and provides an implementation that is incorporated in TinyOS [68]. RI-MAC extends the paradigm with the following features. After data transmission and if the sender has more data packets to send, it uses the acknowledgment beacon as a Ready-to-Receive (RTR) indicator, to start transmitting the next data packet. If there is no incoming data from the sender after transmitting a beacon, the receiver enters the sleep state. The beacon frame in RI-MAC plays a dual role. It is used both as a RTR, broadcasting the request to initiate data transmission, in essence, creating a timeslot for rendezvous, and as an Acknowledgment (ACK), which informs the sender that the data has been received successfully. An optional destination address field is used in the ACK reply to signify a unicast transmission, so that other nodes waiting for a beacon can ignore it. The duty cycle of the beacon transmissions are controlled by varying the sleep state, L, of the node. To prevent coincidental synchronization, a node sets the sleep period randomly between $0.5L$ and $1.5L$, before entering the active state. This essentially makes the average duty cycle of RI-MAC static. An overview of the communication in RI-MAC is shown in Figure 2.2.

If two or more senders contend for the same base beacon, the data packets will be transmitted simultaneously. The experiments conducted in RI-MAC have shown that, due to the presence of the capture effect [128] in FM radios (also called co-channel interference tolerance), such a contending scenario does not necessarily lead to collisions. This property demonstrates that the traditional assumption that a packet collision always results in data corruption is false. For this reason, senders in RI-MAC immediately transmit the data upon receiving a base beacon, without any backoff. The receiver listens for a short period of time after transmitting the beacon, known as the *dwell time*, which is determined by the current backoff window size. Concurrently, it measures the channel power level and processes the bit pattern received. If a valid data frame

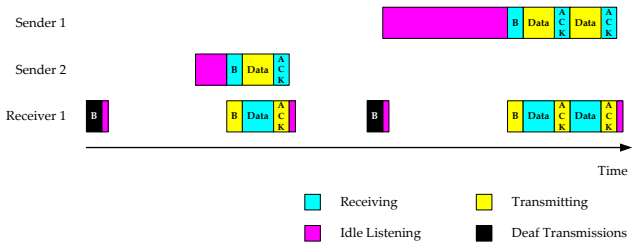


Figure 2.2: Mechanics of RI-MAC, the protocol that made the receiver-initiated paradigm popular. Beacons are sent out by the receiver in order to communicate its availability to receive data.

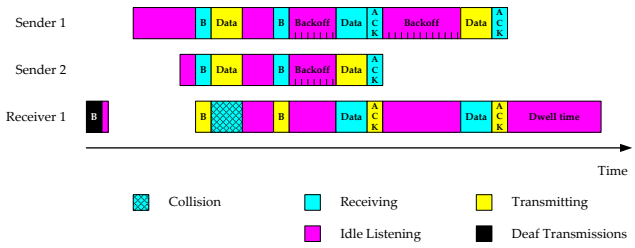


Figure 2.3: Collision avoidance mechanism in RI-MAC, a form of binary exponential backoff.

header is not detected in time, and the measured power level indicates that a transmission is in progress, then, this condition is classified as a collision. Figure 2.3 shows the collision avoidance technique used by RI-MAC. If a collision occurs, the receiver performs a *Clear Channel Assessment (CCA)*, waiting for the channel to be free. Once a clear channel is determined, the receiver transmits a beacon with a backoff window specified, informing the senders of the failed transmission. The senders, that are waiting for an ACK, use the backoff window specified in the beacon to perform a random backoff. The senders listen to the channel, while waiting for the random period to expire, before re-transmitting the data. If a transmission from another sender is detected, the sender withholds the transmission, and waits for an ACK beacon, before resuming with a new random backoff. If a collision happens again, the receiver increments the backoff window using a Binary Exponential Backoff (BEB) [55] strategy, until the maximum window size is reached, after which, the senders and the receiver accept a failed transmission and go back to sleep, retrying at a later point in time.

Beacon-on-Request is an optimization feature, defined by RI-MAC, for when the intended receiver is already active, as shown in Figure 2.4. After a CCA, a sender that has

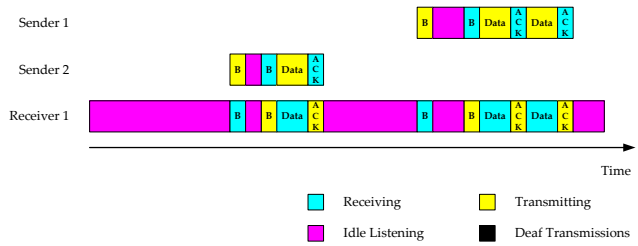


Figure 2.4: Beacon-on-request mechanism in RI-MAC, beacons can be requested explicitly if the intended receiver happens to be awake.

data to transmit, immediately broadcasts a beacon with a backoff window size specified and the destination address set to the intended receiver. The beacon acts as a Ready-to-Send (RTS) indicator, and if the receiver happens to be awake, it replies with a base beacon after a random backoff period. Data exchange then occurs using the normal RI-MAC communication mechanism.

2.3.2.2 Opportunistic Cooperation MAC (OC-MAC) [127]

Opportunistic Cooperation MAC (OC-MAC) [127] extends the beacon-on-request feature to reduce the time that a sender waits for a beacon. Neighboring senders in OC-MAC are allowed to exchange data aggressively while waiting for the receiver to wake up. Figure 2.5 provides an overview of the mechanism used in OC-MAC. Similar to the beacon-on-request feature of RI-MAC, when a node has data ready, it transmits a RTS beacon, if the channel is idle. The beacon contains its residual energy, the destination address, and a request for other senders to relay the data. Notice that, in contrast to the beacon-on-request feature of RI-MAC which is directed towards receivers, the beacon-on-request in OC-MAC is directed only towards senders. By not loading the receivers, this ensures that the channel is not drained of beacons, which would reduce the throughput of the network. After the beacon is transmitted, the sender listens to the channel for a period of time. If it does not receive a response within this duration, the sender looses its right to cooperative communication, and continues to wait silently for a beacon from the receiver or another contending sender.

When an RTS beacon is received by a sender that coincidentally happens to be awake, it compares its residual energy to the contender. The sender ignores the request if the contending sender has more residual energy than itself. If the contender has less residual energy than the sender, it transmits a Clear-to-Send (CTS) beacon, similar to the base beacon in RI-MAC, after a random backoff. The backoff prevents collisions, in case multiple senders are active. The rest of the mechanism is similar to the beacon-

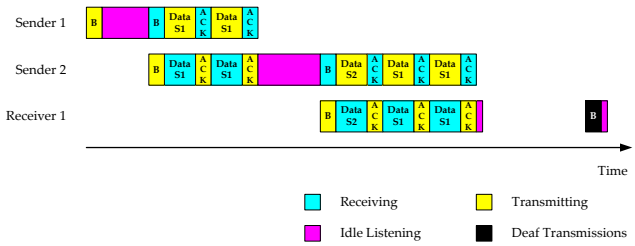


Figure 2.5: OC-MAC extends the beacon-on-request functionality of RI-MAC in a sender-oriented manner.

on-request feature in RI-MAC. Once the exchange of data is completed, the contending sender enters the sleep state, while the sender which received the data, transmits another RTS beacon to check if any opportunity exists to relay both its own data, and the data from the contending sender. Hence, a sender is only permitted to broadcast a RTS beacon immediately after waking up, or after completing a cooperative communication with a contender.

2.3.2.3 Receiver-Centric MAC (RC-MAC) [53]

Receiver-Centric MAC (RC-MAC) is a MAC protocol designed for event-driven applications with heavy traffic loads. It adopts the receiver-initiated paradigm for as long as the network has low traffic for higher efficiency. Differently from RI-MAC where beacon senders transmit immediately upon a beacon reception, RC-MAC requires a initial random backoff in order to increase the fairness between nodes with different transmission power. This approach, on the other hand, is also increasing the energy overhead, since the idle listening is increased. Additionally, in case of collision the senders will retry with a binary exponential backoff whenever the ACK packet has not been received. The receiver is expected to be awake because it just received a frame and it is waiting for a beacon from the next hop. The amount of retries is limited by a predefined number of re-transmission attempts. If this limit is reached, the sender discards the beacon and waits for a new one.

2.3.2.4 Intermittent Receiver-driven Data Transmission (IRDT) [63]

Intermittent Receiver-driven Data Transmission (IRDT) is extending the paradigm with two additional control packets, namely the RACK and the DACK. After the reception of the beacon, named ID, the sender is transmitting the RACK frame to establish the

connection. Then, the data frame transmission follows which is acknowledged with the DACK frame. Additionally, the protocol is defining three collision avoidance mechanisms. The first is CCA with random backoff similar to RI-MAC. The second is based on the frequency of beacon transmissions. The idea is that by increasing the beacons, the senders are stochastically distributed over more beacons and the probability of collision decreases. However, this solution can work only if the receivers are capable of offering their energy resources for forwarding more traffic.

The third collision avoidance mechanism is based on data aggregation. By aggregating multiple data packets into larger frames, the total amount of attempted transmissions falls; thus, the probability of a collision decreases. However, this approach has a negative impact on the delay of each individual data packet. The authors define two methods of collision avoidance with data aggregation, a static one and a dynamic one. According to the static method, the protocol is using a constant buffer of n packets. The node keeps collecting packets from other nodes and locally generated packets into a buffer. When the buffer is full, it is transmitted as a single MAC frame. According to the dynamic method, a sender with a single packet to transmit is waiting normally for the beacon. While waiting, it periodically transmits its own beacons in order to collect packets from neighbors. When the beacon is received, the sender transmits a single frame with as many packets as it managed to collect during that time.

2.3.2.5 Energy Efficient RI-MAC (EE-RI-MAC) [134]

Energy Efficient RI-MAC (EE-RI-MAC) is an enhancement for RI-MAC, defining another approach to increase the energy efficiency of the senders. In particular, EE-RI-MAC uses a technique inspired by X-MAC [13], where, instead of continuously listening for a beacon, a sender alternates between the active state and sleep state within this duration. Figure 2.6 shows an overview of this approach. In order to further reduce the idle listening, senders enter the sleep state after listening to the channel for a period W, and wake up after a duration S. The authors of EE-RI-MAC, opted to use simulations to determine the optimal duty cycle for alternating between the active and sleep state during the idle listening period. It was found that the duty cycle of 37.5%, resulted in the optimum case, outperforming RI-MAC in terms of energy usage. The choice of the value used in the two important parameters, W and S, determines the performance of the scheme. Additionally, even though EE-RI-MAC achieves the same throughput as RI-MAC with higher energy efficiency, the latency of the network suffers.

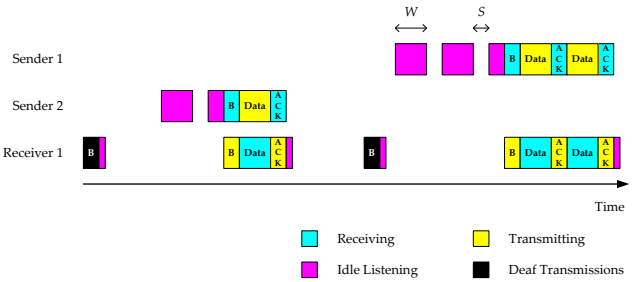


Figure 2.6: EE-RI-MAC introduces the use of duty cycled waiting for beacons in order to reduce idle listening.

2.3.2.6 A-MAC [30]

The key extension of A-MAC to the receiver-initiated paradigm is an extra control packet that aims to reduce the time that a receiver waits for a sender to reply after a beacon transmission. In particular, in A-MAC, the beacon is acknowledged by a short packet named *HACK*. The purpose of this acknowledgment is to quickly inform the receiver of the existence of pending traffic. If the beacon does not trigger a *HACK* packet, the receiver goes directly to sleep. As a result, the receiver wastes less energy in idle listening after each unanswered beacon. In case different *HACK* packets from multiple senders collide, the receiver is still able to assess that there is pending traffic and keeps the radio on. Furthermore, A-MAC incorporates the LPP [86] mechanism for asynchronous network wake-up from deep sleep. In case of no traffic, the network can fall in a deep sleep where the nodes just wake up to transmit beacons very infrequently. Upon an event that should trigger a network wake-up, a node turns on and keeps its radio enabled, listening for beacons. These beacons are answered to with wake-up requests. Nodes that receive such request will propagate it, progressively awaking the whole network. The maximum time required for an asynchronous network wake-up depends on the beacon frequency of the nodes in deep sleep.

2.3.2.7 Routing-Enhanced Asynchronous MAC (REA-MAC) [111]

Routing-Enhanced Asynchronous MAC (REA-MAC) builds on the receiver-initiated paradigm by coordinating the beacon transmissions. The proposed mechanism uses the distance in number of hops of each node from the sink, which is a cross-layer information from the routing layer, to form an *operation cycle*. This cycle is a network-level duty cycle that is built on top of the duty cycles of individual nodes. If N is the maximum distance (in hops) of a node from the sink, the *operation cycle* is split into N

wake-up timeslots. Instead of transmitting beacons independently, each node transmits during the timeslot which corresponds to its particular distance to the sink. Therefore, the beacon transmissions in a network are coordinated to form a multi-hop path like a pipeline and the waiting time in each hop is significantly reduced. Furthermore, a node that has generated data, can keep the radio off during the irrelevant frames to save additional power. The proposed idea is compared to RI-MAC and the simulations show significant reduction of the delivery latency and the power consumption.

2.3.3 Wake-up Prediction

Idle listening constitutes by far the most prevalent source of energy consumption in a receiver-initiated MAC scheme [36]. Several protocols work towards mitigating the time a sender is waiting for a beacon by predicting the next wake-up of the intended receiver.

2.3.3.1 Wide-band MAC (WideMAC) [99]

Wide-band MAC (WideMAC) assumes globally known and static duty cycles, i.e. beacon periods, which are used to predict the next wake-up and decrease the idle listening overhead. In particular, at the beginning a node operates similarly to RI-MAC. Once a node has received a beacon from a receiver node, it predicts the time of the next beacon transmission of the specific node by using the globally known beacon period. Due to clock drifts, the value of this prediction decreases over time, up to a point where it is not longer useful. Whenever a node receives a beacon, it also updates this information.

2.3.3.2 Pseudo-Random Asynchronous Duty Cycle MAC (Pseudo) [67]

In this work, the authors are using a hash function to create pseudo-random wake-up intervals that are uniformly distributed in the range of $[T_{\text{mean}} - T_{\text{range}}/2, T_{\text{mean}} + T_{\text{range}}/2]$, where T_{mean} is the average long term wake-up interval (i.e. the average duty cycle) and T_{range} defines the range of the randomization. Such a randomization, distributes the frame transmissions in the dimension of time, thus decreasing the collisions. Moreover, the hash function is globally known by all the nodes. Thus, each node is able to estimate the next wake-up time of each receiver. Additionally, the authors consider that potential channel contention may introduce delays that can affect the predictions. So, the beacon is enriched with a sequence number and the difference between the wake-up time and the start time of the base transmission. The receiver of the beacon is using the beacon sequence number as input to the hash function in order to predict the next

wake-up time. Then, this prediction is corrected by adding the aforementioned delay. Lastly, each sender wakes up some time before the calculated wake-up time of the receiver, to account for clock drifts. This time is calculated based on the upper bounds of clock drift, given in the datasheets of the microcontrollers.

2.3.3.3 Receiver Wake-up MAC (RW-MAC) [131]

The energy wasted during the idle listening period of the sender is significantly reduced by predicting the wake-up time of the receiver in Receiver Wake-up MAC (RW-MAC). The sender uses the remaining sleep time T_{interval} of a receiver, which is piggybacked on the beacon, to estimate it's wake-up time. Each node maintains a table with the previous time t_{prev} a beacon *should* be received from its neighbors. Initially the sender has to remain awake for a period of time to populate the neighbor table. A sender with data to transmit wakes up after extending the sleep state by the sleep wait time T_{wait} and listens for a beacon from the receiver. T_{wait} is calculated by taking into account the worst case frequency drift θ of the quartz crystal, the static duty cycle T_{cycle} of nodes, and t_{prev}. The maximum time the sender listens to the channel after waking up is set to T_{cycle}, beyond which the node is considered offline or not in the neighborhood.

The beacon and data transmissions are prone to collisions due to the lack of CCA. RW-MAC introduces a stagger wake-up concept as a collision avoidance mechanism. When a sender is initially powered up, it listens to the channel for two consecutive cycles in order to find the maximum gap between two received beacons. It then calculates a non-optimal stagger wake-up offset T_{offset}, based on the midpoint of the gap and T_{cycle}, which is used to permanently shift the beacon cycles of the node. The experimental results show that RW-MAC outperforms RI-MAC for high traffic loads. It supports a higher number of concurrent data flows and consumes less energy than its counterparts due to its low duty cycle.

2.3.3.4 Predictive Wake-up MAC (PW-MAC) [113]

Predictive Wake-up MAC (PW-MAC), is a receiver-initiated scheme that reduces the energy consumption of senders, inspired by WiseMAC [32]. PW-MAC, uses an independently generated pseudo-random sequence for controlling the wake-up times of each node, allowing senders to accurately predict the time when a receiver will wake up, similarly to [67]. An on-demand prediction error correction mechanism helps to compensate for timing challenges caused by unpredictable hardware, operating system delays, and clock drifts. Furthermore, the predictable wake-up times are used to improve the performance in case of collisions and channel errors. In case there is need for a retransmission, senders in RI-MAC stay awake until receivers wake up again. On

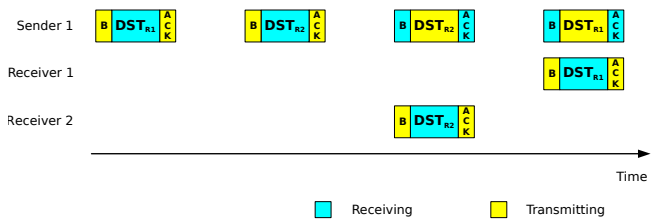


Figure 2.7: Frame reordering in RP-MAC, frames are sent out according to the beacon interleaving pattern.

the contrary, senders in PW-MAC wake up at the next predicted receiver wake-up time, minimizing the energy spent waiting for the receiver.

2.3.3.5 Reordering Passive MAC (RP-MAC) [51]

Reordering Passive MAC (RP-MAC) extends RW-MAC with a feature called *Frame Reordering (FR)*. The FR scheme reduces the delivery latency by using the next wake-up information of several receivers to reorder the transmission buffer of the sender. For instance, consider the scenario depicted in Figure 2.7, where the buffer of the sender has a frame for $R1$ that is followed by a frame for $R2$. However, the next wake-up of $R2$ will happen before the next wake-up of $R1$. The glsfr scheme reorders the two frames to significantly reduce the waiting time. Compared to RI-MAC, RP-MAC achieves better energy efficiency and lower end-to-end delay.

2.3.4 Adaptive Duty Cycling

Dynamic adaptation of the sleeping schedules is optimizing the performance of the paradigm to given dynamic conditions. Dynamic duty cycling can be based on several parameters such as the topological structure, the traffic conditions or the energy input.

2.3.4.1 Stair-like Sleep Asynchronous RI MAC (Stair) [124]

Receiver-initiated sensor networks suffer from the fundamental limitation that the energy that a node spends waiting for a beacon, depends on the availability of the receiver node to receive traffic. Therefore, a low duty cycling receiver will force the transmitter to waste a significant amount of energy, leading to sub-optimal network performance. To make matters worse, the closer a node is to the sink, the more network traffic it has

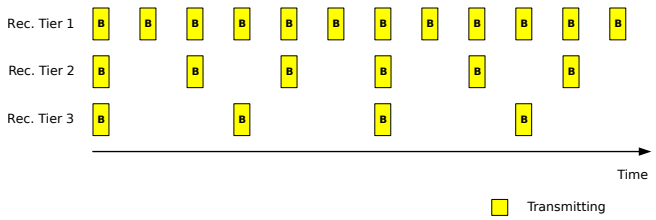


Figure 2.8: Stair-like beaconing pattern determined by a node's tier (lower numbers mean shorter distance from the sink).

to serve. The authors propose an asynchronous receiver-initiated protocol that builds upon this limitation. In particular, the authors show via simulations that the overall network performance, in terms of packet delivery ratio, packet delay and energy efficiency, can be significantly improved by adapting the duty cycles considering the number of hops of each node from the sink. Such an adaptation would lead to stair-like sleeping pattern (Figure 2.8), in which the closer a node is to the sink the more time it stays active. Despite the promising results at a network level, the individual node's energy capability to support the higher duty cycles should be taken into consideration. Furthermore, it is interesting to note that the same beneficiary effects would result from a topology designed with more nodes placed closer to the sink.

2.3.4.2 On Demand MAC (ODMAC) [38]

ODMAC builds upon the foundation of the receiver-initiated paradigm for the realization of EH-WSNs, which are sensor networks that are powered by energy that is harvested from the surrounding environment. ODMAC uses an adaptive duty cycle mechanism based on the ENO principle [58], where the energy consumed by a node is less than or equal to the amount of energy harvested. All nodes in the network dynamically adjust the beacon and sensing duty cycle, in order to achieve and maintain an ENO-Max state [120], which is defined as an ENO state with maximum performance. This means that when the node is consuming more energy than it harvests, the duty cycles are decreased to reduce the energy consumption. In the same manner, when the energy consumed is lower than the energy harvested, the duty cycles are increased so that the node is more active. Nodes in the network have the dual role of being a receiver for forwarding tasks and sender for measuring tasks. ODMAC decouples the duty cycles of these two roles in a single node. Hence, a node has a beacon duty cycle and a sensing duty cycle. The beacon duty cycle controls the trade-off between energy consumption and end-to-end delay, while the sensing duty cycle controls the trade-off between energy consumption and throughput. Therefore, ODMAC gives to an administrator the ability to decide the trade-offs depending on the application.

Moreover, ODMAC defines a forwarding policy based on opportunity. Instead of waiting for the intended receiver to wake up, a sender opportunistically forwards data using the first available beacon that leads towards the desired destination. Since the probability of receiving beacons from a receiver with surplus energy is high, this policy creates a more robust network, that is adaptive to changes in energy, by maintaining a balanced load in the network. Furthermore, the idle listening time of senders is reduced in the region where the receivers' coverage overlaps.

In addition to random backoff, ODMAC also includes a novel low-overhead collision avoidance mechanism, named AB [41], that detects potential collisions and avoids them before the beacon transmission. Additionally, AB can provide QoS by prioritizing urgent traffic. Lastly, ODMAC includes a security protocol for the authentication of receiver, that is surveyed in Section 2.3.8.

ODMAC and its features are presented in detail and evaluated in Chapter 3 and its following chapters.

2.3.4.3 Self Adapting RI-MAC (SARI-MAC) [65]

Self Adapting RI-MAC (SARI-MAC) self-adapts to the traffic load by adjusting the beaconing frequency to the estimated traffic. In particular, the maximum duration between two beacons is capped by the maximum link delay that is allowed by the application. Moreover, the duration between two beacons is also adapted so that the average beaconing rate is equal to the average traffic rate. The later adaptation ensures that the beacon transmission frequency is large enough to serve the incoming traffic. SARI-MAC also introduces a novel collision avoidance mechanism through time slot reservation. After the beacon transmission, a contention window period follows during which, the nodes pick a uniformly random slot to request for a timeslot reservation. At the end of the contention window, the receiver sends back to all the contending nodes a report with the reservations. Nodes transmit their data during their reserved timeslot, which is long enough for a data packet and the respective acknowledgment.

2.3.5 Quality of Service (QoS)

The protocols that focus on QoS provide services that prioritize the traffic according to the needs of the overlaying application.

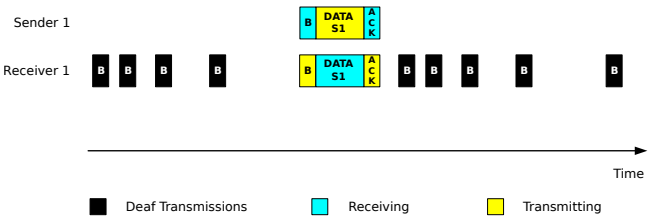


Figure 2.9: Traffic dependent beaconing pattern as shown in CyMAC, beacons become sparser over time whenever there is no sender to serve, and reset as soon as a new one is found.

2.3.5.1 Delay Bounded MAC (CyMAC) [92]

Delay Bounded MAC (CyMAC) focuses on delay-sensitive applications and attempts to provide data delivery guarantees. This builds upon a unique feature introduced by CyMAC. In CyMAC, the beacons are dedicated for each neighboring sender. Thus, the period of each individual beacon can be independently adapted on a per-sender basis. The conduced comparison with RI-MAC suggests that CyMAC can provide delay guarantees under various traffic conditions. Except for cases of tight required delay bounds, CyMAC yields lower duty cycles than RI-MAC.

The protocol also introduces a dynamic duty cycle adaptation mechanism that aims to adjust the sleeping schedules to the given traffic conditions. Thus, when the traffic is light, sensor nodes sleep more and conserve more energy, while when the traffic is heavy, they broadcast more beacons to increase the performance. The duty cycle adaptation algorithm operates as follows. All nodes operate at a maximum duty cycle and as long as they don't serve any traffic they exponentially increase the time between two beacons. The exponential increase continues until a data packet arrives, which triggers the node to reset the duty cycle period to its minimum value. This scheme is shown in Figure 2.9.

2.3.5.2 QoS Aware Energy-Efficient MAC (QAEE-MAC) [61]

QoS Aware Energy-Efficient MAC (QAEE-MAC) extends the receiver-initiated paradigm with a mechanism that allows priority data to be transmitted faster than normal data. Upon waking up, each sender transmits a control packet, named *Tx-beacon*, which indicates the priority of its data packet. Before the beacon transmission, the receiver wakes up and collects *Tx-beacon* packets. Then, it uses the priority information to determine to which node to transmit to. However, such support for priority packets comes at the cost of extending the idle listening time of all the involved senders.

2.3.6 Broadcast Support

In asynchronous duty cycling sensor networks, broadcasting constitutes a challenge because nodes are not awake concurrently. For applications and protocols that require broadcasting services, MAC protocols have been enriched with mechanisms to support them.

2.3.6.1 Asynchronous Duty cycle Broadcasting (ADB) [106]

Asynchronous Duty cycle Broadcasting (ADB) extends RI-MAC with support for broadcasting. Similarly to unicasting, broadcasting is initiated by the receiver. Therefore, the procedure is equivalent to a series of unicast transmissions. ADB avoids transmissions over poor links, by entrusting the packet that needs to be broadcasted to other nodes. The sender tracks the procedure by maintaining two lists of neighboring nodes (those who received the broadcasted packet and those who are assigned to other nodes) and goes to sleep once all its neighbors are marked in either of these lists. Consider the example that a sender S wants to broadcast a frame to $R1$ and $R2$ and assume that the quality of the link between S and $R2$ is poor, while the link between $R1$ and $R2$ is good. After the transmission of the packet from S to $R1$, the receiver $R1$ takes the responsibility of forwarding the packet to $R2$. The coordination of the procedure, which includes the information of which nodes are pending and the quality of the respective links, is achieved by control data that is piggy-backed on the beacons and data frames.

2.3.6.2 Yet Another MAC (YA-MAC) [130]

In Yet Another MAC (YA-MAC), the nodes go through an initialization phase in which all nodes are on 100% duty cycles. During this phase, they determine their neighborhood and agree on some protocol parameters. One of these parameters is the broadcast time interval, which defines the period of a broadcast slot. All nodes wake up during the broadcast slot, which makes normal broadcasting feasible. The nodes are loosely synchronized. In particular, the Synchronization Error Tolerance Window (SETW) defines a guard time interval that protects the system from minor clock drifts. If the synchronization falls below a desired level, nodes are triggered to enter an 100% duty cycle phase during which synchronization is re-established. Lastly, YA-MAC uses the amount of neighboring nodes, as it is determined in the initialization phase, to select the contention window for collision avoidance.

2.3.6.3 Receiver Wake-up Broadcast (RWB) [96]

Receiver Wake-up Broadcast (RWB) extends RW-MAC [131] with broadcast support. Similarly to ADB [106], a broadcast transmission consists of a series of unicast transmissions. The key difference to ADB is that packets are not delegated to other nodes. Instead, RWB uses the wake-up prediction mechanism of RW-MAC to optimize the performance. Moreover, the individual unicast transmissions that compose a broadcast transmission can be optionally acknowledged to optimize the delivery ratio.

2.3.7 Multi-Channel Extensions

Exploiting multiple channels increases the capacity of a link. Hence, it can lead to higher throughput, fewer collisions and shorter delays in networks with relatively high traffic. On the other hand, overlapping WSNs that are using multi-channel MAC protocols, interfere with each other, as they cannot be tuned to different orthogonal channels. Wireless sensor nodes are also typically limited by a single radio unit. As a result, MAC protocols cannot operate at multiple channels concurrently in order to transmit and receive in parallel. A series of multi-channel MAC protocols that are using the receiver-initiated paradigm are surveyed next.

2.3.7.1 Duty Cycle Multi-channel MAC (DCM) [69]

Duty Cycle Multi-channel MAC (DCM) defines three types of channels, namely a single Control Channel (CC), a series of data channels and a single Broadcast Channel (BC). Normal unicast communication is executed as follows (Figure 2.10). A sender that wants to transmit is actively listening to the CC for incoming beacons, named *Announcement (ANC)*. When ready to receive, the receiver transmits an ANC on the CC. The ANC frame includes the number of a data channels which is selected by the receiver randomly. The authors claim that due to duty cycling and the single-radio limitation, random channel selection is a better choice than information-based selection. Right after the ANC transmission, the receiver is switching to the selected data channel and listening for a RTS frame. Right after the reception of the expected ANC, the sender also switches to the announced channel. The communication then follows a typical RTS - CTS - DATA - ACK communication. Random Backoff (RB) is also included for avoiding collisions between multiple nodes that received the same ANC. If a node finds the CC or the specified data channel busy for multiple times, it assumes that the network is congested and goes back to sleep. Moreover, a sender that is not receiving an ANC for a predetermined period of time is transmitting an ANC in CC in order to avoid deadlocks.

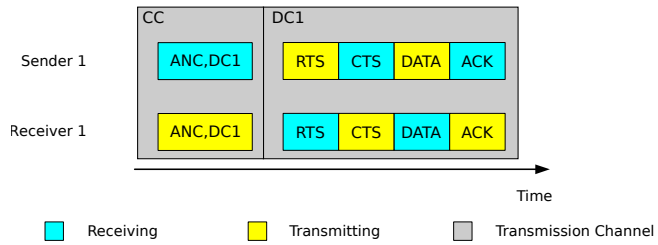


Figure 2.10: Multi-channel approach in DCM. A control channel (*CC*) is used to choose a specific data channel (*DCx*) where the communication will be carried through.

DCM also provides multi-channel broadcast support via the BC. Whenever a sender wants to broadcast a frame, it switches to the broadcast channel and after a CCA it transmits the broadcast frame for M consecutive time intervals. Every node, no matter how the duty cycles are configured, has to switch to the BC and check for possible incoming broadcast data once every $M - 1$ time intervals. The value of M can control the trade-off between energy efficiency and broadcast latency.

Asynchronous Receiver-initiated Multi-channel MAC (ARM) [70] constitutes a follow-up publication by DCM's main authors and it extends its analysis and evaluation. However, there are no significant changes in the core of the protocol. ARM operates similarly to DCM.

2.3.7.2 Efficient Multi-channel MAC (EM-MAC) [112]

Efficient Multi-channel MAC (EM-MAC) is a multi-channel MAC protocol that does not use a common control channel as the channel numbers and wake-up schedules are not explicitly exchanged. Instead, every node generates a channel number and a time for the next wake-up event using a shared pseudo-random number generator. Every node is able to predict the next wake-up event of any other node just by knowing the *prediction state*. The prediction state includes the information of the random seed, a previous wake-up time, a multiplier a and a constant c. A node that does not have the prediction state of a given receiver, listens for a beacon on the first channel, which contains the corresponding information. Additionally, each node maintains the status of each channel by counting when CCA fails. If the status metric exceeds a certain threshold the channel is blacklisted and is not used. If the pseudo-random number generator chooses a blacklisted channel, the node stays on the previous channel. Blacklisted channels are advertised using a bitmap on the beacons.

The rest of the protocol's operation is based on the receiver-initiated paradigm. Differ-

ent from RI-MAC, EM-MAC puts sender to sleep if the collision resolution mechanism does not resolve the collision before the receiver goes back to sleep. The ability to predict the next wake-up through the pseudo-random generator, allows the node to sleep and save energy in the meantime.

2.3.8 Security

TinySec [60] is a security suite for WSNs that provides important services such as data integrity and confidentiality at link level. TinySec is fully compatible with the receiver-initiated paradigm. However, TinySec cannot protect receiver-initiated MAC protocols from beacon replay attacks. A replay attack is defined as an attack against a protocol where previously exchanged messages are reused in order to fool legitimate participants into thinking that the current run of the protocol is valid and exchanged data is fresh [24]. Beacons contain the identity of their creator which is the main piece of information needed to determine whether or not a specific beacon can be used by a potential sender, according to the overlying routing algorithm. By replaying beacons, it is possible to deploy a series of other attacks.

2.3.8.1 Receiver Authentication Protocol (RAP) [25]

RAP is a challenge-response authentication protocol that is included inside ODMAC and aims to authenticate the receiver, i.e. the beacon transmitter, in a receiver-initiated data transmission. It has two modes of operation, namely *detection* and *prevention* mode. The detection mode, Receiver Authentication Protocol - Detection (RAP-D), is a low overhead scheme and aims at detecting an intruder that replays beacons without preventing it from doing so. The prevention mode, Receiver Authentication Protocol - Prevention (RAP-P), on the other hand, is a more costly scheme that prevents the attack altogether.

RAP is presented in detail in Chapter 3 and evaluated in Chapter 6.

2.4 Reflection

All the protocols surveyed in Section 2.3 define mechanisms and features that can be added to the basic paradigm to optimize its performance. It should be noted that such features can be used in different combinations beyond the definition of each individual

protocol. Depending on the properties of a specific application a network administrator can combine features, introduced by different protocols, to optimize the overall performance of the system. Sensor networks are mainly characterized by the limited resources of its nodes. A holistic network design is vital for the efficient use of the limited resources. The MAC protocol, as a fundamental part of the networking stack, should be configured with respect to the topological structure of the network, the power source of the nodes and the characteristics and requirements of the running application.

A key design decision is between static and adaptive duty cycles, as many of the presented features are not compatible with both. Adaptive duty cycles are expected to be beneficial only in dynamic network conditions, as they would introduce overhead otherwise. The energy profile of the nodes, which is the combination of the energy input and energy consumption profile, plays a key role. When the energy profile of the nodes of the system is unbalanced, static duty cycles would introduce bottlenecks in the network. A balanced energy consumption profile implies a carefully designed static topology and stable traffic generation, in such a way that the duties of all nodes are balanced. A balanced energy input profile implies that the nodes are powered by batteries with similar energy resources. In this case, significant energy can be saved by predicting the upcoming wake-up using a backup prediction scheme that assumes static duty cycles, like WideMAC [99], Pseudo [67] and PW-MAC [113]. If there are no other networks deployed in the same area, multiple channels can further increase the performance (EM-MAC [112]).

In the opposite case, e.g. dynamic topologies, applications with bursty traffic or nodes that are powered by unpredictable energy that is harvested from the environment, a dynamic duty cycle approach is recommended. In addition to using the specific adaptive duty cycle features when relevant (Stair [124], CyMAC [92], ODMAC [38] and SARI-MAC [65]), idle listening can be reduced either by predicting the next wake-up using the approach of RW-MAC [131], by using the multiple receivers as described by the opportunistic forwarding mechanism of ODMAC [38] or by using a the duty cycled listening approach of EE-RI-MAC [134]. Moreover, the use of multiple channels is feasible using the approach of DCM [69], which is compatible with dynamic duty cycles.

Independent of how the duty cycling is organized, the beacon acknowledgment proposed by A-MAC [30] mitigates the cost of beaconing. In case any form of wake-up prediction mechanism is used, this information can be used to optimize the transmission buffer, as the frame reordering feature of RP-MAC [51] defines. If, on the other hand, no wake-up prediction mechanism is used, the operation cycles of REA-MAC [111] reduce the idle listening, while the opportunistic cooperation, proposed by OC-MAC [127], and the altruistic backoff of ODMAC [38] handle collisions in a way that also mitigates idle listening. Otherwise, BEB, as described in RI-MAC [107] or RC-MAC [53] can be used. Such methods constitute active collision avoidance mechanisms. In cases of very low traffic, random access via random beacon selection (e.g.

IRDT [63]), would sufficiently handle collisions without the additional overhead.

The rest of the features provide services for the application or protocols at a higher level and, therefore, should only be used if these services are needed and the network is capable of handling the additional overhead. The approach of QAEE-MAC [61] and CyMAC [92] can be used for traffic differentiation and applications with priority requirements. TinySec [60] and RAP [25] can be used for applications with security requirements. For broadcast support, the approach of RWB [96] can be used along a wake-up prediction mechanism, while ADB [106] or YA-MAC [130] can be used otherwise.

Tables 2.2 and 2.3 present all the surveyed protocols in a more compact way. More specifically, Table 2.2 provides a top-down approach, where each protocol is described and characterized in terms of its implemented features. On the other hand, Table 2.3 uses a complementary bottom-up organization, showing what technique is used to address each challenge and by which protocols it is implemented.

2.5 Conclusions of the Survey

In this chapter we have surveyed all the receiver-initiated MAC protocols for WSNs, classifying them according to their different properties. The main goal of the survey is to provide the reader with enough insight into each protocol so that further review of the relevant literature can be carried out autonomously.

As briefly discussed in Section 2.4, there is no global solution that performs well in every possible environment and application. On the contrary, a specific technique could be very good in one scenario and disastrous in another. Alongside this, a strong integration and a tight interaction between the different components of a protocol, again dictated by the needs introduced by the overlying application, are vital for the achievement of a successful solution. Under these assumptions, a protocol designer is supposed to mix and match the presented protocol features in order to craft a solution that perfectly suits the needs of the desired application.

Table 2.2: The list of features that each protocol implements in a $Protocol \rightarrow Features$ classification.

Protocol name	Features summary
A-MAC	Idle listening minimization, Collision avoidance
ADB	Broadcast
CyMAC	Adaptive D/C, QoS
DCM	Multiple channels, Broadcast
EE-RI-MAC	Idle listening minimization
EM-MAC	Wake-up prediction, Multiple channels
IRDT	Collision avoidance
OC-MAC	Idle listening minimization Cross-layer interaction, Collision avoidance
ODMAC	Adaptive D/C, Idle listening minimization Cross-layer interaction, Collision avoidance
PW-MAC	Wake-up prediction
Pseudo	Wake-up prediction
QAEE-MAC	QoS, Idle listening minimization Cross-layer interaction, Collision avoidance
RAP	Cross-layer interaction, Security
RC-MAC	Collision avoidance
REA-MAC	Cross-layer interaction Idle listening minimization
RI-MAC	Collision avoidance
RICER	Wake-up prediction, Cross-layer interaction
RP-MAC	Frame reordering, Collision avoidance
RW-MAC	Wake-up prediction Idle listening minimization, Collision avoidance
RWB	Broadcast
SARI-MAC	Adaptive D/C, Cross-layer interaction, Collision avoidance
Stair	Adaptive D/C
WideMAC	Wake-up prediction, Idle listening minimization Collision avoidance
YA-MAC	Collision avoidance, Broadcast

Table 2.3: A specific challenge is addressed by each protocol in different ways. The table provides a $Challenge \rightarrow Protocols$ approach, a complementary view to Table 2.2.

Challenge	Technique	Protocols
Idle listening	Wake-up prediction	EM-MAC, PW-MAC Pseudo, RICER,RP-MAC RW-MAC, WideMAC
	Beacon acknowledgment	A-MAC
	Listening duty cycle	EE-RI-MAC, QAEE-MAC
	Cross-layer interaction	OC-MAC, ODMAC REA-MAC, RICER
	Beacon period adaptation	IRDT, SARI-MAC
	Indirect	IRDT, ODMAC
Collision avoidance	Random backoff	A-MAC, DCM, EM-MAC IRDT, OC-MAC, ODMAC RC-MAC, RI-MAC RICER, QAEE-MAC WideMAC, YA-MAC
	Cooperation	OC-MAC, ODMAC
	Data aggregation	IRDT
	Beacon period adaptation	IRDT,SARI-MAC
	Timeslot reservation	SARI-MAC
	Staggering	RW-MAC
	Multi-channel extensions	DCM, EM-MAC
Adaptive duty cycling	Traffic based	CyMAC, SARI-MAC
	Energy based	ODMAC
	Distance based	Stair
Quality of service	Frame reordering	CyMAC, QAEE-MAC
Broadcast	Synchronization	DCM, YA-MAC
	Multiple unicasts	ADB, RWB
Security	Authentication	RAP

CHAPTER 3

The ODMAC Protocol

3.1 A Receiver-Initiated MAC Protocol for EH-WSNs

ODMAC is an experimental MAC protocol that has been designed specifically for EH-WSNs, in an attempt to satisfy the two key system goals of EH-WSNs: sustainability and application performance (see Section 1.2.2).

ODMAC follows the receiver-initiated paradigm of communication between duty-cycling nodes. This design direction is justified as follows. A key requirement of MAC schemes for EH-WSNs is the ability to independently adjust the duty cycle of an individual node to adapt to the energy the node can harvest. Therefore, the synchronous paradigm is considered unsuitable for EH-WSNs as, in a synchronous network, the duty cycles of the sensor nodes are coupled to each other via a global clock. Furthermore, asynchronous schemes have been shown to be more energy-efficient that synchronous approaches [48, 95]. Within the asynchronous approach, the receiver-initiated scheme is shown to be more energy efficient [71, 107] than the sender-initiated schemes. To verify the related work, Section 7.2 includes an analytical comparison between the two asynchronous paradigms in the context of EH-WSNs.

The purpose of ODMAC is not to provide full MAC layer functionality covering all the elements a MAC protocol is meant to address. Instead, the goal of ODMAC is to be a testing platform that would allow the experimentation and evaluation of dif-

ferent features that aim to provide adaptability and improve the energy-efficiency of EH-WSNs.

The unpredictable, ever-changing and small-scale nature of the energy input makes adaptable radio duty-cycling the only means to achieve sustainable operation. The duty cycles need to adapt to energy input of different orders of magnitude. Specifically, in energy constrained environments, the MAC protocol needs to support very low duty cycles in order to guarantee the long-term sustainability of the system. On the other hand, when the energy is abundant, the same protocol has to efficiently use the energy surplus to increase the application performance. Beyond the adaptable duty cycles, ODMAC incorporates additional features that address most of the challenges of receiver-initiated MAC protocols (see Section 2.2), including the mitigation of idle listening, the energy-efficient avoidance of collisions, the provision of QoS via differentiation of high-priority traffic and the provision of means to securely authenticate the origin of a beacon.

The key features of ODMAC are: adaptive duty cycles (Section 3.2), opportunistic forwarding (Section 3.3), collision avoidance and traffic differentiation with AB (Section 3.4) and the RAP (Section 3.5). The remaining features of ODMAC are summarized in Section 3.6.

3.2 Basic Operation and Adaptive Duty Cycles

The receiver-initiated paradigm constitutes the foundation of all the receiver-initiated asynchronous protocols, including ODMAC. According to the paradigm, a node willing to receive data, wakes up periodically and checks for incoming transmissions. To do so, a CCA is performed immediately after waking up, and a special message called *beacon* is broadcasted only if the channel is free and afterwards the receiver continues to listen to the channel for a short predetermined period of time. Meanwhile, whenever a node with data ready to be sent enters the active state, it listens silently for a beacon from the intended receiver. Once the beacon is received, the sender transmits its data packet, and waits for another beacon which acknowledges (ACK) the reception of the data. Conversely, if there is no incoming data after transmitting the beacon, the receiver enters the sleeping state. At this point both the sender and receiver resume their cycles normally.

As a MAC scheme specifically targeted for EH-WSNs, ODMAC builds upon the receiver-initiated paradigm, shown in Fig. 3.1. To adapt in the ever-changing unpredictable nature of the energy input, nodes dynamically adjust their duty cycle in a completely independent and distributed manner. Nodes in the network have a double role of *receivers* for forwarding tasks and *senders* for measuring tasks. ODMAC decouples the

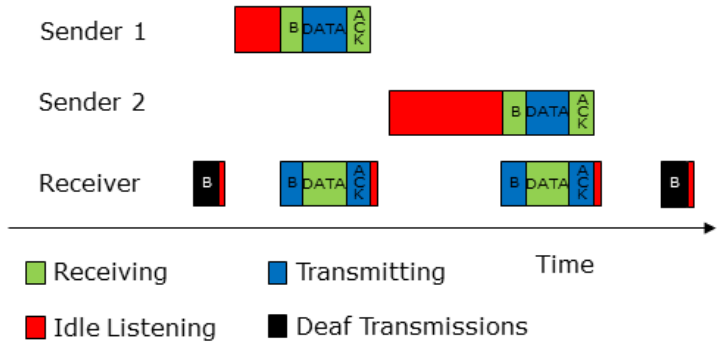


Figure 3.1: Mechanics of ODMAC.

duty cycles of these two jobs within a single node. Hence, a node has a *beaconing duty cycle* and a *sensing duty cycle*. The beaconing duty cycle controls the trade-off between energy consumption and end-to-end delay, while the sensing duty cycle controls the trade-off between energy consumption and throughput. Thus, ODMAC grants the network administrator the ability to establish the trade-off depending on the particular application. Moreover, the period of beaconing (t_b) is randomized uniformly within $[t_b - R, t_b + R]$ for random channel access, where R defines the level of randomization.

ODMAC adapts the duty cycles based on the ENO principle [58]. According to the ENO principle, a node is sustainable if, over a time period that its energy buffers can support, the energy consumed is less than or equal to the energy harvested. All nodes in the network dynamically adjust the beacon and sensing duty cycle, in order to achieve and maintain an ENO-Max state [120], which is defined as an ENO state with maximum performance. This means that when the node is consuming more energy than is harvesting, the duty cycles are decreased to reduce the energy consumption. In the same manner, when the consumed energy is lower than the harvested energy, the duty cycles are increased. Thus, the adaptation of the duty cycles follows a greedy approach. In practice, the level of the energy buffer is being monitored and the duty cycles are changed periodically towards the desired operation point.

3.3 Opportunistic Forwarding

Opportunistic Forwarding is a forwarding feature that exploits the random nature of a beacon reception towards the energy-efficiency and sustainability of the network. Typically, MAC and routing functionalities are implemented into different layers of abstraction, namely the link and network layer respectively. The routing protocol has

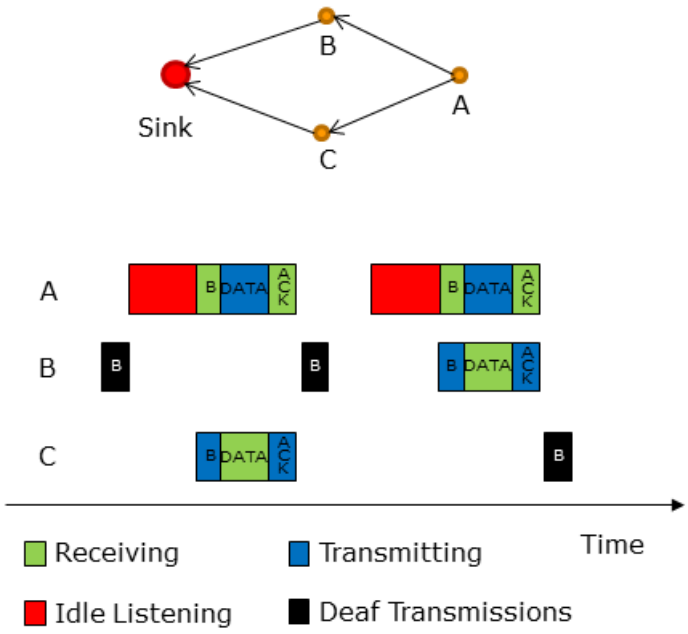


Figure 3.2: Example of opportunistic forwarding in ODMAC.

the duty to identify the next receiver (i.e. the next hop) based on the optimum path (see Section 1.3.5). The identity of the receiver is, then, fed to the MAC protocol, which has the duty to find the receiver within the broadcasting domain and transfer the data to it. The operation of the two protocols is repeated for every link until the sink is reached.

If a routing protocol is aware that the energy consumption (i.e. the idle listening while waiting for a beacon) depends on the duty-cycles, it can include this information in its routing metric and, essentially, route traffic, more energy-efficiently, through the nodes that have higher beaconing frequencies. This solution has two limitations. The first limitation is that the selected node is overloaded with all data transmissions. The second limitation is that routing traffic through the nodes that transmit beacons more frequently does not always minimize the time a node waits for a beacon. Consider the following motivating example, as illustrated in Figure 3.2 (up). The sensor node A has two routing options (node B or C) to reach the sink node. Assuming that node B transmits beacons more frequently, the routing protocol selects node B as the receiver. On average, the beacons from the selected receiver will be received sooner and less energy will be consumed in idle listening compared to the alternative option. Yet, if we evaluate each packet transmission separately, there will be some rare cases that a beacon from node C would arrive earlier than node B.

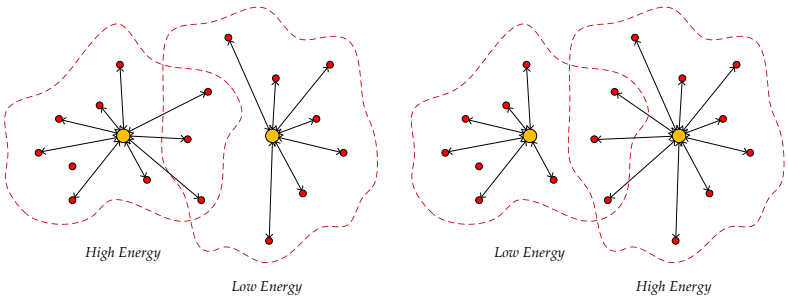


Figure 3.3: Opportunistic forwarding in ODMAC in a multi-sink, single-hop network.

Opportunistic forwarding builds upon these two limitations. Instead of waiting for a specific receiver to wake up, a sender opportunistically forwards data to any approved receiver (anycast routing), based on the beacon obtained *first*, as illustrated in Figure 3.2. Since the probability of receiving beacons from a receiver with surplus energy is high, this policy creates a more robust network, that is adaptive to changes in energy, by keeping the load in the network balanced between the routing options. In the long-term, the traffic is divided, in a fully autonomous manner, to multiple receivers according to the harvested energy and their duty cycles (Figure 3.3). Inherently, the traffic distribution autonomously adapts to changes in the energy input, as it follows the adaptation of the duty cycle. Furthermore, this mechanism significantly improves the energy-efficiency of the system, as the time senders spend waiting for a beacon (i.e. idle listening), and therefore their energy consumption, is reduced.

Opportunistic forwarding requires a routing protocol that assigns each sender a list of approved receivers. Existing routing metrics are applicable. The only required change is that the routing protocol needs to feed ODMAC with the identities of the $n \geq 1$ best receivers. In Section 3.6, we provide a simple hop-count routing protocol that is compatible with opportunistic forwarding.

3.4 Altruistic Backoff (AB)

In receiver-initiated MAC protocols, beacons form time slots of communication. Randomization techniques can distribute data transmissions among multiple beacons. Nevertheless, when multiple nodes wake up and wait for the same beacon, a collision is inevitable. Unless there are specific conditions that allow receivers to provide the network with much more beacons than the generated data packets, receiver-initiated protocols are particularly vulnerable to collisions.

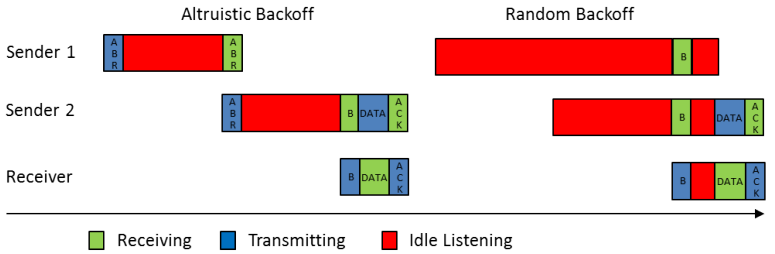


Figure 3.4: Collision avoidance with Altruistic Backoff (AB) and Random Backoff (RB). In RB the inevitable collision is resolved after the beacon transmission, while both nodes waste energy in idle listening waiting for it. AB uses control packets (ABR) to resolve the inevitable collision before the beacon allowing the nodes to back off earlier and save energy by decreasing the time they spend in idle listening.

The standard solution for collision avoidance is named RB. The idea is that the MAC protocol defines a time interval (*timeslot*) and a Contention Window (CW). Before transmitting, each node selects a random number, chosen uniformly between zero and $CW-1$, and it delays the data transmission by that number of timeslots, while listening to the channel for other transmissions. If the channel remains idle, data transmission follows. If the channel gets occupied by another transmission, the node backs off and attempts to transmit at a later time. Variations of the RB algorithm are the most commonly used collision avoidance mechanisms in receiver-initiated MAC protocols (see Table 2.2 in Chapter 2).

The mechanics of RB imply that senders that contend for the same beacon will spend a vast amount of energy waiting for the beacon and the collision will be detected and resolved only after the beacon transmission. AB is a collision avoidance mechanism that detects potential collisions and avoids them before the actual beacon transmission. Specifically, a node with data to transmit wakes up and, before starts waiting for a beacon, it transmits a control packet, named Altruistic Backoff Request (ABR), that identifies the beacon(s) the node is waiting for. A node that is already waiting for the same beacon and receives this packet altruistically backs off, offering the beacon to the node that wakes up last. At the low overhead of one extra control packet transmission per data transmission, collisions are mitigated and idle listening is significantly reduced. Figure 3.4 shows an example of collision avoidance with AB compared to RB, that provides intuition on the benefits of the former.

The presented collision avoidance scheme does not suffer from fairness issues for two reasons. First, WSNs consist of cooperative nodes that do not have incentives to overutilize the channel. Furthermore, random channel access provides similar probabilities for all nodes to use the beacon. Essentially, the beacon and thus the channel is taken

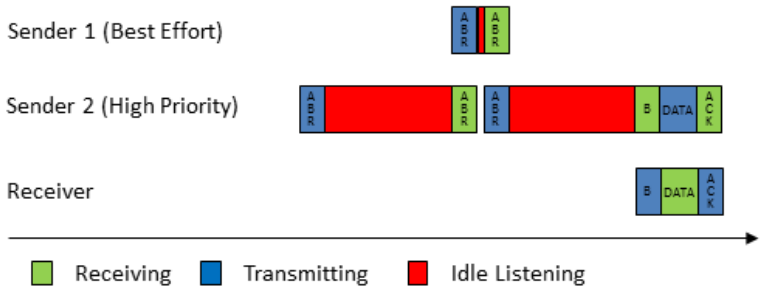


Figure 3.5: Traffic differentiation with Altruistic Backoff (AB). Nodes with traffic of high priority, upon being silenced by nodes with lower priority, immediately retransmit an ABR to retake the beacon.

by the sender that wakes up last. Therefore, random channel access guarantees long-term fairness. In other words, as long as different senders have equal opportunities to wake up last, they have equal opportunities to take the beacon. Similarly to RB, long-term fairness can be compromised if nodes do not follow the protocol. In particular, if a node continuously retransmits an ABR, it will always get the beacon. Generally, we do not consider this a problem, because WSNs are networks of cooperative nodes that do not have incentives to favor their performance against the performance of other nodes. However, this property is a security vulnerability that can lead to DoS attacks. Nevertheless, security protocols, such as RAP (see the Section 3.5), can be used to authenticate control packets in an energy-efficient manner and secure the protocol against such attacks.

Beyond being a security vulnerability, this property is used for QoS services through traffic differentiation. Traffic differentiation is valuable in case of applications that generate traffic of different urgency (e.g. alerts vs. monitoring traffic). We define two types of data packets that correspond to two traffic classes, the high-priority class and best-effort class. The priority number that defines the priority class is included in the ABR. Upon the reception of an ABR, a node compares the priority number indicated in the ABR to the priority number of the local packet it has to transmit. If and only if the local packet belongs to the high-priority traffic class and the remote packet belongs the best-effort traffic class, the node immediately transmits a new ABR to retake the beacon, as shown in Figure 3.5. As a result, the priority number guarantees that ABR retransmissions occur only when a node has a higher priority than the node who currently has the beacon.

Upon a backoff event, the time of a next transmission attempt can follow different policies with respect to the importance of the data. We can consider two extremes. On one hand, the sender might attempt to transmit immediately, as recommended for

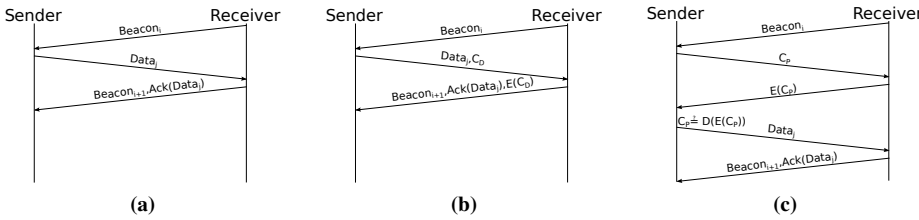


Figure 3.6: Basic Receiver-Initiated Communication (a), RAP-D (b), RAP-P (c).

traffic of high priority. On the other hand, the sender might choose to buffer the packet and transmit it together with the following packet. We recommend this policy for best-effort traffic, as it is the policy that minimizes the energy consumption. Additionally, the sender might choose a solution in between that compromises the advantages and the disadvantages of the two extremes. Unless stated otherwise, we assume the use of the second policy.

3.5 Receiver Authentication Protocol (RAP)

Receiver-initiated MAC protocols are particularly vulnerable to beacon replay attacks. Even encrypted and authenticated beacons can be captured and replicated by an attacker to attract traffic with the end-goal of either selective or network-wide DoS attacks. ODMAC incorporates a security protocol that countermeasures this vulnerability. RAP [25] is a challenge-response authentication protocol that aims to authenticate the receiver, i.e. the beacon transmitter, in a receiver-initiated data transmission. RAP is compatible and can be used on top of every MAC protocol that follows the receiver-initiated paradigm, essentially securing the whole class of protocols from beacon replay attacks; moreover, it can and should be used together with security suites that provide other important features such as data integrity and confidentiality (e.g. TinySec [60]).

RAP has two modes of operation as shown in Fig. 3.6, namely detection and prevention mode. In a nutshell, the detection mode (RAP-D) is a low overhead scheme and aims at detecting an intruder that replays beacons without preventing it from doing so. The prevention mode (RAP-P), on the other hand, is a more costly scheme that prevents the attack altogether. The key difference between the two modes is the timing of the challenge-response message exchange. In RAP-P, the challenge-response message exchange takes place *before* the data transmission. Thus, the sender transmits the data packet only if the receiver is authenticated. The low overhead nature of RAP-D, on the other hand, is maintained by piggybacking the challenge and its response on top of the frames normally exchanged in the MAC protocol. In other words, the authentication of

the receiver takes place *after* the data transmission (thus, the attack is not prevented). Having energy-efficiency as a primary system priority, the idea is that a node normally operates at the low overhead detection mode and switches to the expensive prevention mode only if necessary.

RAP-D is aiming at detecting beacon replay attacks with low communication overhead. The protocol works as shown in Fig. 3.6b. Consider that a sender node A wants to transmit some data to a receiver node B. After B broadcasts a beacon, A answers back with a data packet and a challenge value C_D. On its following beacon, B acknowledges the reception of the data packet, and attaches the encrypted version of the challenge $E_{k_{RAP}}(C_D)$ using the protocol specific shared key k_{RAP}. At this point B can validate the response to the challenge by decrypting it and checking it against its original value. Should these two values not match, then B can conclude that the initial beacon was not genuine. RAP-D adds a minimal overhead in the whole communication scheme, as the challenge and the response are piggybacked on top of a regular message exchange. Furthermore, if the challenge, C_D, is transmitted as part of the payload and encrypted with it, its size can be relatively small without risking increasing the chances of a space exhaustion attack.

RAP-P is aiming to prevent the beacon replay attack at the cost of an increased overhead. In particular, the challenge-response messages are exchanged before the data transmission, in order to distinguish the legitimate from the replayed beacons. The protocol works as shown in Fig. 3.6c. Instead of sending the data right after a beacon, A sends out a longer challenge C_P, and awaits for its encrypted version $E_{k_{RAP}}(C_P)$ from B. Only if the received value decrypts correctly (i.e. matches against C_P), then data is sent. This scheme is more expensive because it requires two additional messages to be exchanged. Additionally, the size of the challenge needs to be significantly larger than the detection mode to prevent space exhaustion attacks.

Depending on the security goal of an application, RAP can be configured to switch between the two modes, using several transition policies. If the application cannot tolerate a few beacons getting replayed, the protocol should always operate in prevention mode for maximum security. In the opposite case, the detection mode should be the default mode to promote energy-efficiency. Here, the transition from RAP-D to RAP-P is done after a defined number of challenge mismatches. This number should be tuned accordingly to account for channel errors. The transition back to detection mode is done either automatically or manually depending on the level of desired of security. In cases of high security requirements, it may be desired that RAP-D is reactivated manually by the system administrator only after an investigation. Alternatively, an automatic transition to RAP-D is performed after a predetermined number of successful challenge matches. To avoid the exploitation of the latter transition policy, this number is exponentially increased each time a new replay attack is detected.

3.6 The Remaining Features

Contrary to the aforementioned key features, this section presents the less-contributing features of ODMAC that are based either on the direct application of state-of-the-art schemes or on simple solutions that are yet to be investigated, extended and optimized.

3.6.1 Loose Binding Mode (LBM)

In cases of extremely low power conditions, in which the receiver is transmitting beacons at a very low frequency, the sender can select to operate in *Loose Binding Mode (LBM)*. LBM operates as follows. Each node includes in the beacons its current beaconing period. The sender uses this information to loosely bind with the receiver and adjust the sleeping period accordingly. This approach reduces the idle listening at the cost of additional delays. The practical design and implementation of LBM can be based on RW-MAC [131].

3.6.2 Command & Control Channel

In order to minimize the energy consumption, protocols from all communication layers should aggregate data and minimize packet transmissions. To achieve this goal, ODMAC incorporates in the control packets of the MAC layer a delay-tolerant *Command & Control* channel. When this channel is used by other protocols of the system or the application, no additional packets need to be generated. Specifically, after the data transmission, the receiver transmits back an extended beacon, named *Command & Control Beacon (CCB)*. Apart from acknowledging the data reception, the packet may include piggybacked data from other protocols. As an example, the Command & Control channel can be used by an application to change the configuration of the nodes.

3.6.3 Link-Layer Authentication and Encryption

Security extensions have been included within ODMAC to provide confidentiality and integrity. The security subsystem is loosely based on TinySec [60] and provides four modes of operations: *no security*, *authentication*, *encryption*, *authentication+encryption*. All the properties are provided by using the same inexpensive cryptographic primitive which currently is either Skipjack [1] or PRESENT [12], both in Cipher-Block Chaining (CBC) mode [105]. This guarantees good performance at a minimal overhead. According to the required functionality, authentication and encryption can be activated

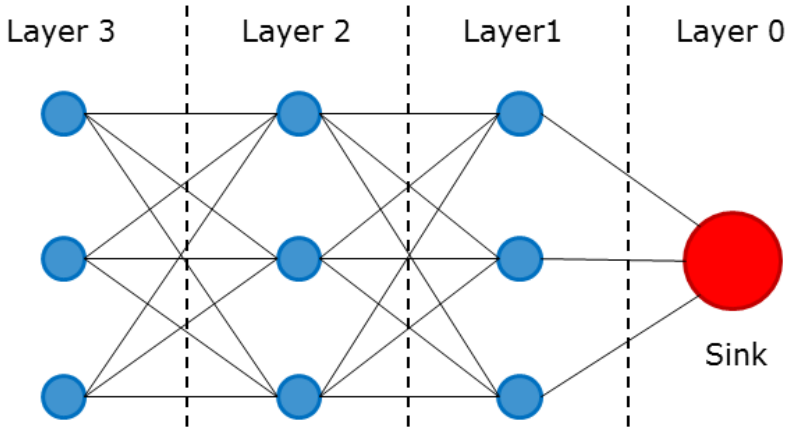


Figure 3.7: Layer-based Anycast Routing (LAR) routes the traffic to *any* node that is one layer closer to the sink.

with a single message granularity. Besides encrypting and authenticating payload messages, it is also possible to do the same for beacon messages.

3.6.4 Layer-based Anycast Routing (LAR)

Layer-based Anycast Routing (LAR) is a simple, minimal overhead, hop-count routing protocol that selects multiple forwarders and, therefore, is compatible with opportunistic forwarding. While, technically, not part of ODMAC and the MAC layer, we chose to present it here, as it can be implemented inside the MAC layer in whole.

The scheme operates as shown in Figure 3.7. We define $layer(u)$ as the distance of node u form the sink, expressed in number of hops. The sink is initialized at layer 0. All nodes advertise their layer through their beacons and nodes update their layer upon beacon reception. Let B be the set of layers received by node u then $layer(u) := \min(B) + 1$. Additionally, layers are reset ($layer(u) = \infty$) if no beacon is received after a predefined amount of time. The candidates for receivers are considered the nodes advertising a layer lower than the one of the sender, thus leading towards the sink. More formally, a sender u forwards a frame to node v if and only if $layer(v) < layer(u)$. By using the beacons to distribute information required for routing decisions, we avoid transmitting extra control packets and save energy. Moreover, the routing scheme is resilient to nodes entering and exiting the network.

3.7 Protocol Evaluation Summary

Following Chapters 4-8 aim to evaluate ODMAC and its features using mathematical analysis, simulations and testbed experiments. In particular, Chapter 4 evaluates the features of adaptive duty cycles (Section 3.2) and opportunistic forwarding (Section 3.3) using analysis and OPNET simulations. Chapter 5 evaluates AB (Section 3.4) using MATLAB simulations. Chapter 6 formally verifies and evaluates RAP (Section 3.5). Chapter 7 analytically compares ODMAC with two widely used MAC protocols. Lastly, Chapter 8 experimentally evaluates ODMAC in a testbed of eZ430-rf2500 wireless sensor nodes.

CHAPTER 4

Adaptive Duty Cycles and Opportunistic Forwarding

4.1 Evaluation Overview

In this chapter, we focus on the evaluation of ODMAC with respect to the features of Adaptive Duty Cycles (Section 3.2) and Opportunistic Forwarding (Section 3.3). In Section 4.2, we provide initial intuition on the beneficial properties of Opportunistic Forwarding in a single link. Section 4.3 models a multi-hop EH-WSNs and Section 4.4 continues the analysis in a multi-hop context. The analytical results are, then, supported by simulations in OPNET [88]. Section 4.5 presents the implementation of the protocol as a process in OPNET and Section 4.6 presents the simulation results. Lastly, Section 4.7 summarizes the evaluation.

4.2 Analysis of Opportunistic Forwarding

We start the analysis by providing some initial intuition about the beneficial properties of Opportunistic Forwarding in a single link. Suppose that a sender i has to transmit one frame. Assuming an overlaying routing protocol that provides n_i candidate nodes

as potential receivers (e.g. LAR, Section 3.6.4). The sender may forward the frame to one of these receivers. Each one of the candidates has a beaconing period, t_j where j identifies the candidate receiver.

4.2.1 Modeling the Expected Waiting-for-a-Beacon Delay

First, we model the expected time a node spends in idle listening waiting for a beacon from any one of the forwarding candidates. Let X_j be the waiting time for the beacon of node j. Also let x_j be the expected value of X_j. We define as *waiting-for-a-beacon* delay (Y_i) the time node i spends until it receives a beacon from *any* of the n_i forwarding candidates. Let y_i be the expected value of Y_i. By definition, the following equation is true.

$$P(Y_i \leq y_i) = 0.5 \tag{4.1}$$

The probability that none of forwarding candidates transmits a beacon in less than or equal to y_i is given by (4.2).

$$\prod_{j=1}^{n_i} P(X_j > y_i) \tag{4.2}$$

Therefore the following statement is also true.

$$P(Y_i \leq y_i) = 1 - \prod_{j=1}^{n_i} P(X_j > y_i) \tag{4.3}$$

Assuming random channel access, X_i follows a uniform distribution. Therefore, its probability can be estimated as follows.

$$P(X_j > y_i) = \frac{t_j - y_i}{t_j} \tag{4.4}$$

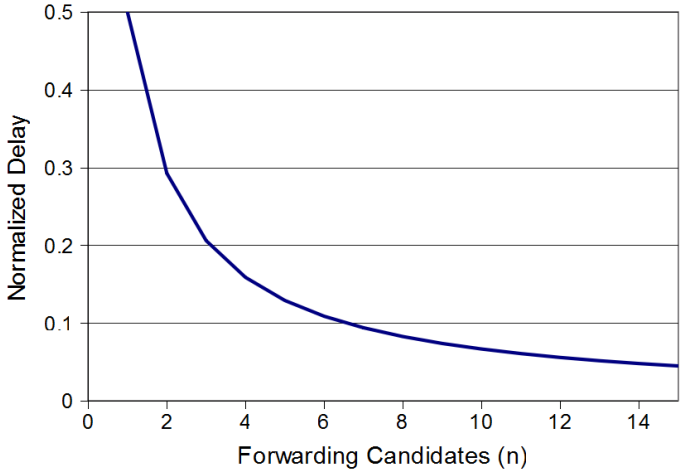


Figure 4.1: Expected waiting-for-a-beacon delay (y_i) normalized to the beaconing period t.

From the statements expressed in (4.1), (4.3) and (4.4), we derive to the equation (4.5).

$$\prod_{j=1}^{n_i} \frac{t_j - y_i}{t_j} = 0.5 \tag{4.5}$$

Equation (4.5) is a n_i-th degree polynomial equation that estimates the expected waiting-for-a-beacon delay (y_i) in seconds, which is the expected time node i spends in idle listening waiting for a beacon from any of the receivers in its candidate list.

4.2.2 Intuition on Opportunistic Forwarding

Next, we simplify equation (4.5) to provide some initial intuition about the beneficial properties of Opportunistic Forwarding. Let's assume that all nodes have the same beaconing period, $t_j \equiv t$.

$$y_i = t(1 - 0.5^{\frac{1}{n_i}}) \tag{4.6}$$

Equation (4.6) gives us an estimation of the expected waiting-for-a-beacon delay (y_i). Figure 4.1 plots the estimated value of (y_i) normalized to the beaconing period t, for

various values of forwarding candidates (n_i). Observe that y_i is decreasing exponentially as the number of forwarding candidates increase. This improvement becomes less significant for higher values of n_i. The highest value is when there is just one forwarding candidate. Essentially, this case is equivalent to unicast routing.

Figure 4.1 indicates that increasing the forwarding candidates exponentially decreases the time a node waits for a beacon. The benefits are twofold. Both the energy consumption and the measurement delivery delay decrease. Hence, Opportunistic Forwarding contributes in both the energy-efficiency and the application performance of the WSN.

Moreover, a performance trade-off arises. Since (4.6) is monotonous, increasing the number of forwarding candidates always improve the performance of the MAC layer. However, from the perspective of the routing layer, increasing the number of forwarding candidates also increases the use of suboptimal routing paths. Therefore, there is room for cross-layer optimization.

4.3 Modeling multi-hop EH-WSNs

Let us now zoom out of a single link and model an entire network. We consider a multi-hop WSN with a single sink node. Each sensor node i generates traffic periodically at a sensing rate of s_i.

4.3.1 Node-to-Sink Delay

The node-to-sink delay is composed by the sum of every link delay in each intermediate hop. The link delay consists of five components. We consider significant only two of the components, namely the transmission delay and the synchronization delay. The processing delay is the time a microprocessor spends processing the data packet and is generally considered negligible in comparison to the other sources of delay. The propagation delay is also considered negligible as it depends on the speed of light and the links in sensor networks are relatively short. The queuing delay is also considered insignificant as they system generates and forwards very low amounts of data. The transmission delay is equal to $(L * 8)/R$, where L is the packet size in bytes and R is the transmission rate of the link in bits per second. The synchronization delay is estimated by solving equation (4.5) for y_i. The sum of those gives us the link delay (d_i) for node i.

$$d_i^l = \frac{L * 8}{R} + y_i \tag{4.7}$$

We cannot trivially calculate the node-to-sink delay by adding the delay of all the traversed links, because a different path is used for every packet due to Opportunistic Forwarding. Hence, the node-to-sink delay needs to be probabilistically modeled.

Remember that an overlaying routing algorithm provides the MAC layer with a set of n_i forwarding candidates. The probability that a packet generated by i will be forwarded by node j is given by $p_{i,j}$ where the sum iterates over the nodes that are in the list of appropriate forwarders, n_i.

$$p_{i,j} = \frac{1}{t_j \sum_{a=1}^{n_i} \frac{1}{t_a}} \tag{4.8}$$

The node-to-sink delay (d_i^s) in sensor i is equal to the local link delay (d_i^l) plus the respective node-to-sink delay of each potential forwarder with respect to the probability of it being the actual forwarder. This is given by the following equation where the sum iterates over the nodes that are in the list of appropriate forwarders, n_i.

$$d_i^s = d_i^l + \sum_{j=1}^{n_i} p_{i,j} d_j^s \quad \text{[sec]} \tag{4.9}$$

The node-to-sink delay of each node can be calculated by propagating the estimated delay backwards, i.e. from the sink node towards the sensor nodes of the outer layer. For the nodes that have direct access to the sink, Eq. (4.9) still applies with $p_{i,sink} = 1$, $d_{sink}^s = 0$ and $d_i^l = (L * 8)/R$.

4.3.2 Traffic Rate

The total traffic that a sensor transmits (r_i) consists of the traffic it generates by sensing (r_i^g) and the traffic it forwards on behalf of other nodes (r_i^f). The traffic rate generated locally is equal to $r_i^g = 1/s_i$, where s_i is the period of the sensing duty cycle. In addition to that, every backwards neighbor contributes with a part of its total traffic rate with respect to the probability of node i being the actual forwarder (given by (4.8)). The latter is given by the following equation where the sum iterates over the nodes that have node i in their list of forwarding candidates, m_i.

$$r_i^f = \sum_{k=1}^{m_i} p_{k,i} r_k \quad \text{[pkt/sec]} \tag{4.10}$$

The total traffic that a sensor transmits (r_i) is estimated by the following formula.

$$r_i = r_i^g + r_i^f = \frac{1}{s_i} + \sum_{k=1}^{m_i} p_{k,i} r_k \quad \text{[pkt/sec]} \tag{4.11}$$

The traffic rate of each node can be calculated by propagating the estimated traffic rate forwards, i.e. from the sensor nodes of the outer layer towards the sink node. For the nodes that are in the outer layer of the network, Eq. (4.11) still applies with $m_i = 0$.

4.3.3 Power Consumption and Generation

Next, we model the power consumed in communication, which is the most significant source of power consumption. During the operation of a sensor node, the radio is changing between idle, transmitting and receiving modes. Hence, the instantaneous power consumption changes over time. The power consumption model presented in this section considers the long-term average power consumption. Essentially, the instantaneous power consumption is replaced by its long-term equivalent constant power consumption.

The long-term average power consumed for transmitting packets (P_i^{ttx}) is given by (4.12) where r_i is given by (4.11), the ratio of the packet size (L) over the transmission rate (R) is the duration of the transmission and P_i^t is the power consumed while transmitting.

$$P_i^{ttx} = P_i^t r_i \frac{L * 8}{R} \quad \text{[W]} \tag{4.12}$$

For the value of P_i^t, we use the power consumption model presented in [125]. In particular, the power consumed for transmission is given by the following formula where P_i^{tx} is the selected power of the transmitted signal, η is the drain efficiency and P_{t0} is the constant power consumed in the circuits of the radio module.

$$P_i^t = P_{t0} + \frac{P_i^{tx}}{\eta} \quad \text{[W]} \tag{4.13}$$

The long-term average power consumed for receiving packets (P_i^{trx}) is given by the following formula where P_{r0} is the power consumed when the radio is in receiving

mode, r_i^f is the traffic rate of the forwarded packets and the ratio of the packet size (L) over the transmission rate (R) is the time required for the reception.

$$P_i^{trx} = P_{r0} r_i^f \frac{L * 8}{R} \quad \text{[W]} \tag{4.14}$$

The long-term average power consumed while waiting for a beacon (P_i^w) is estimated by the following formula where P_{r0} is the power consumed when the radio is in receiving mode, y_i is the waiting time given by (4.5) and r_i is given by (4.11). In the exceptional case of LBM, approximately no power is consumed, $P_i^w \approx 0$.

$$P_i^w = P_{r0} y_i r_i \quad \text{[W]} \tag{4.15}$$

Lastly, the long-term average power consumed for beaconing (P_i^b) is given by the following formula where t_i is the beaconing period, the ratio of the beacon size (L_b) over the transmission rate (R) is the time required for a beacon transmission and P_t is the power consumed while transmitting.

$$P_i^b = P_i^t \frac{1}{t_i} \frac{L_b * 8}{R} \quad \text{[W]} \tag{4.16}$$

The sum of all the aforementioned sources of energy consumption give the overall long-term average power consumption of node i.

$$P_i^{tot} = P_i^{ttx} + P_i^{trx} + P_i^w + P_i^b \quad \text{[W]} \tag{4.17}$$

The long-term average power generated by the energy harvester, P_i^{in}, is modeled as a random variable that follows a normal distribution with a mean of μ and an variance of σ^2.

HCR_i, which is defined as the ratio of P_i^{in} over P_i^{tot} (given by (4.17)), gives us the operating state of the node. Whenever $HCR_i > 1$, node i operates at a sustainable state (i.e. ENO). If the ratio is in $[1, 1.1]$, we consider the node to operate at a sustainable state with maximized performance (i.e. ENO-Max).

$$HCR_i = \frac{P_i^{in}}{P_i^{tot}} \tag{4.18}$$

4.3.4 Transmission Range

The transmission range model is based on the link budget formula. P_i^{rx} is signal's power at the receiver in dBm, P_i^{tx} is the power of the transmitted signal in dBm, G^{tx} and G^{rx} are the antenna gains at the transmitter and receiver in dBi, respectively, and PL_i is the signal attenuation over the path, i.e. path loss, in dB. We consider the antenna gains to be the same at all nodes, $G^{tx} = G^{rx} = G$.

$$P_i^{rx} = P_i^{tx} + G^{tx} + G^{rx} - PL_i \quad \text{[dBm]} \tag{4.19}$$

The path loss at a distance d_i is given by the following equation, where e is the loss exponent.

$$PL_i = P_1 + 10\log(d_i^e) \quad \text{[dBm]} \tag{4.20}$$

P_1 is the path loss in the first meter (dB) assuming free space model, where f is the frequency of the signal (MHz).

$$P_1 = 20\log(f) - 27.55 \quad \text{[dB]} \tag{4.21}$$

If we equate P_i^{rx} to the receiver's sensitivity threshold, d_i becomes the transmission range of node i and is estimated by solving the equations (4.19), (4.20) and (4.21) for d_i.

4.4 Analytical Evaluation

This section uses the model presented in Section 4.3 to evaluate the effectiveness of ODMAC to promote the sustainability and the application performance of an Energy Harvesting - Wireless Sensor Network (EH-WSN) through adaptive duty cycling and opportunistic forwarding.

4.4.1 Model Configuration for Analytical Experiments

The presented formulae (Section 4.3) effectively model a multi-hop EH-WSN. Given an arbitrary set of nodes with either positions in $A \times A$ field, and a set of input param-

eters for each one of them, we first estimate the transmission range, which then defines the topology. Given a set of forwarding candidates for each node, we calculate the performance metrics that are in our interest, such as the harvested-to-consumed long-term average power ratio and the node-to-sink delay.

For the analysis we assume the use of LAR (see Section 3.6.4). Hence, the set of forwarding candidates for each node is set to all the sensor nodes that are one hop closer to the sink node. Moreover, we select the transmission power of each node aiming to maximize the number of links between the nodes. In particular, we select the maximum supported transmission power and then we gradually decrease it to the point that no links are broken. Table 4.1 provides the values of the remaining parameters of the model. These values apply to all sensor nodes and suppose using the CC1000 transceiver [125].

Table 4.1: Model parameters.

L	100 Bytes	G	0 dBi	P^{tx}	10 dBm
L_b	8 Bytes	e	4	η	0.157
R	256 Kbps	P^{rx}	-96 dBm	P^{t0}	15.9 mW
f	433 MHz	A	300 m	P^{r0}	22.2 mW

We consider three different energy harvesting conditions. The long-term average power input, P_i^{in}, of each node is a random variable that follows a normal distribution with the respective parameters as summarized in Table 4.2. The three scenarios cover a large variety of energy harvesters according to [102].

Table 4.2: Energy harvesting conditions.

Name	Mean	Variance
EH1	1 mW	0.2
EH2	0.3 mW	0.05
EH3	0.1 mW	0.02

Lastly, we consider a random topology of 50 nodes, unless stated otherwise. Based on these parameters, the transmission range is approximately 105 meters. The sink node is placed in position $(0, 0)$, leading to a 5-hop deep topology. Further experiments in different random topologies verify the same trends.

4.4.2 Intuition on Adaptive Duty Cycles

Increasing the beacon period (t_i) has two opposite effects in the energy consumption of the network. From one side, the long-term average power consumption due to beacon-

ing is decreased. On the other side, the nodes that depend on the node's beacon need to spend more time waiting for a beacon, wasting energy in idle listening. The following experiment suggests that there is a threshold above which it is not beneficial to increase the beaconing period for saving energy.

In this experiment, we simplify the model in order to provide some initial intuition about the adaptive duty cycling. Let's assume that all nodes have the same beaconing period, $t_i \equiv t$, and sensing period, $s_i \equiv s$. Figure 4.2 shows the long-term average power consumption of the sensor nodes for different maximum beaconing periods. Different lines represent a different sensing periods, s, in seconds. Observe the minimum that derives from the aforementioned trade-off. The minimum gradually increases as the sensing period increases. We can fit these optimum values in an inverted exponential function.

$$t = 2.26 \cdot e^{\frac{-354}{s+155}} \tag{4.22}$$

The experiment indicates that it inefficient to set the beaconing period at higher values than the minimum (t_{max}). Thus, the system has the following operating alternatives. The system can trade power for shorter delays if t_i is adapted in $(0, t_{max}]$. The system can also trade power for throughput by adapting s_i. Alternatively, the system can operate at the local minimum, t_{max} for a maximum acceptable sensing period, s_i and use any excess of energy elsewhere (e.g. security).

4.4.3 Application-Specific Scenarios

In this section we focus on two classes of WSN applications, namely delay-sensitive applications and offline-analysis applications.

4.4.3.1 Delay-Sensitive Applications

To support delay-sensitive applications, the system should guarantee energy neutral operation and invest the excess of harvested energy in decreasing the delays. We assume that the applications are characterized by a maximum sensing period requirement in seconds, s_{max}. Similarly, a minimum sensing period is defined, s_{min}.

The harvested-to-consumed long-term average power is monitored and the duty cycles are adapted as follows, until the system stabilizes. Initially, all sensors set their sensing period to $s_i := s_{min}$ and their beaconing period to $t_i := t_{max}$, where t_{max} is given by

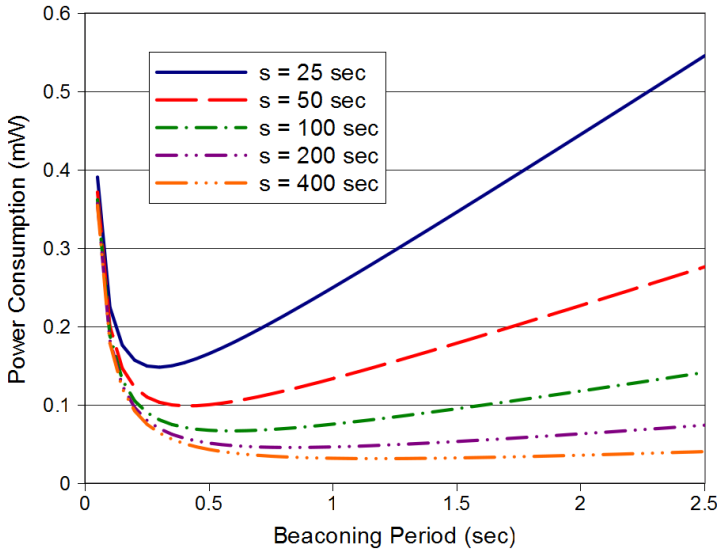


Figure 4.2: Long-term average power consumption for different beaconing (t) and sensing periods (s).

(4.22). If a node has an excess of energy, it decreases the beaconing period. If, on the other hand, a node needs to save energy, it first increases the beaconing period up to the maximum value, t_{max}. If this is not enough to achieve a sustainable state, the sensing period is increased up to its maximum value, s_{max}. If this is still not enough, the node switches to LBM and it binds to the node with the minimum beaconing period.

Table 4.3 shows the results of several numerical experiments on a 50-node random topology after stabilization. The EH-WSN is tested under the three environmental energy conditions given in Table 4.2 and under different application requirements. We consider that $s_{min} = s_{max}/2$. The table shows the average sensing rate of the nodes in packets per minute (*throughput*), the average node-to-sink delay in ms (*delay*) and the average Harvested-to-Consumed power Ratio (HCR). The last column (*sustainable nodes*) gives the number of nodes that operate at a sustainable state (ENO).

Under the *EC1* power input, the system has plenty of energy to operate at the maximum desired sensing rate while any excess of energy is used to decrease the node-to-sink delay as much as possible. The operation of all nodes is sustainable. Under the *EC2* power input, we notice that the average node-to-sink delay is higher, which shows that the system effectively uses the harvested energy to improve the performance metric of interest, namely the node-to-sink delay. Also observe that in the most demanding sensing requirements ($s_{max} = 50$), some of the nodes now need to increase their

Table 4.3: Numerical results for delay-sensitive applications.

P^{in}	s_{max}	throughput (ppm)	delay (ms)	HCR	sustainable nodes
EC1	50	2.4	21.2	1.09	50
EC1	100	1.2	23.1	1.07	50
EC1	200	0.6	21.5	1.09	50
EC2	50	2.36	48.7	1.09	50
EC2	100	1.2	57.1	1.1	50
EC2	200	0.6	48.4	1.09	50
EC3	50	1.57	238.1	1.06	47
EC3	100	0.94	215.1	1.08	49
EC3	200	0.54	187	1.08	50
EC4	3200	0.019	1600	1.09	50

sensing period to achieve a sustainable state, leading to a lower sensing rate compared to *EC1*. As we decrease the energy input further more, the average node-to-sink delay gets higher and the sensor node need to loose the application sensing rate requirements (s_{max}) in order to operate in a sustainable state (*EC3*).

To summarize the key conclusions of this experiment, ODMAC can effectively adapt its energy consumption to different energy inputs of various orders of magnitude, providing sustainable operation. Additionally, we see that that the average node-to-sink delay is decreased as the system it exposed to higher levels of energy. This shows that the harvested energy is used to favor the performance metric that is selected to be the most important.

4.4.3.2 Offline-Analysis Applications

In this subsection, we move our focus to the set of applications where the measurements are going to be analyzed offline. In this class of applications, the sensing rate or throughput is the performance metric of interest.

Similarly to the previous case, the harvested-to-consumed long-term average power is monitored and the duty cycles are adapted as follows, until the system stabilizes. Thus, the sensing period, s_i, of the nodes is adapted with respect to the application requirements (s_{min}, s_{max}). The challenge, in adapting the sensing rate, is that any changes have direct effects in all the intermediate nodes between the node and the sink. Thus, the outer nodes, that have less forwarding tasks, may flood the network with a lot of packets that the inner nodes are unable to handle. Hence, the duty cycles are adapted as follows. If the node needs to save energy and the maximum sensing period

is reached, the node asks for help to its children nodes. For this request we use the command & control channel (see Section 3.6). The nodes that receive this request increase their sensing period to help the parent node. Additionally, they lock their sensing period to its current value and they do not adapt it unless they receive a new help request. The beaconing period, t_i, is set to the value the minimizes the energy consumption given by (4.22).

Table 4.4 shows the results of numerical experiments on a 50 node random topology after stabilization. Similarly to the previous section, we expose the system to the three different levels of environmental energy. We assume that the applications require a minimum sensing rate defined by s_{max}. We also consider that $s_{min} = s_{max}/10$. The table shows the average sensing rate of the nodes in packets per minute (*throughput*), the average node-to-sink delay in ms (*delay*) and the average HCR. The last column (*sustainable nodes*) gives the number of nodes that operate at a sustainable state (ENO).

Table 4.4: Numerical results for offline-analysis applications.

P^{in}	s_{max}	rate (ppm)	delay (ms)	HCR	sustainable nodes
EC1	200	5.9	28	1.09	50
EC2	200	2.65	62	1.09	50
EC3	200	0.69	253	1.08	50

The table shows that in all cases the nodes manage to balance in a sustainable state. Any excess of harvested energy is now used to increase the sensing period. Observe the increasing trend in the average sensing rate as the power input increases. Again, the harvested energy is used to favor the performance metric that is selected to be the most important.

4.4.4 Node Density

Another way to improve the application performance is by increasing the density of the network, i.e. increasing the amount of nodes that cover the desired area. The higher the amount of nodes, the higher the total energy input that the system harvests; energy that can be used to improve the performance. In addition to increasing the throughput of the network, deploying additional nodes is also expected to decrease the average node-to-sink delay. As Figure 4.1 suggests, the expected waiting time for a beacon decreases exponentially as the amount of neighbors increase. This results to an overall decrease of idle listening and the average node-to-sink delay, that can be attributed to Opportunistic Forwarding.

In the next experiment, we expose randomly generated networks of various sizes to *EC3* and let the nodes stabilize their duty cycles using the algorithm for delay-sensitive

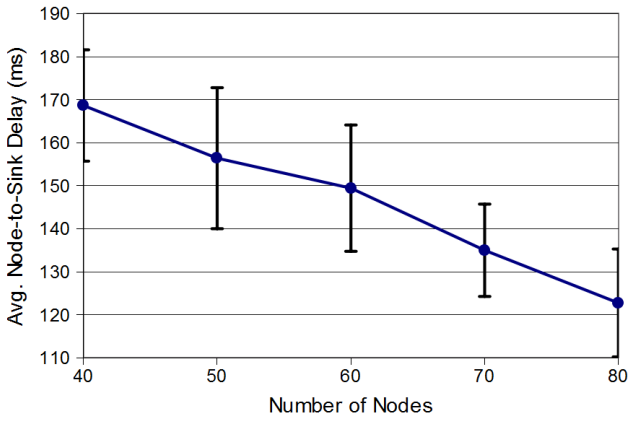


Figure 4.3: Average node-to-sink delay over random topologies of various node densities.

applications. Figure 4.3 shows the average sink-to-node delay of 20 runs and their respective 90% confidence intervals. We observe a clear improvement of the delay as the network density increases.

4.5 Implementation for the OPNET Simulator

To evaluate ODMAC through simulations, we implement the protocol in the OPNET Simulator [88]. OPNET is a proprietary network simulator. By modeling networking protocols in all layers of the communication stack and simulating discrete events, OPNET is able to predict the behavior of complex computer networks. In OPNET, every protocol implemented as a Finite State Machine (FSM).

The implementation of a sensor node and a sink node in OPNET follows a modular approach. In the case of a sensor node, the Application (APP) layer consists of a *Sensor* module that is in charge of simulating the sensing functionality by generating data packets. The generated packets are forwarded to the MAC layer. In the case of a sink node, the APP layer consists of a *Sink* module that simply gathers the data packets from the MAC layer and calculates statistics.

An explicit routing layer does not exist. Instead, the routing functionalities are incorporated inside the MAC layer using the LAR protocol (see Section 3.6.4), which considers that all the nodes that are one hop closer to the sink are forwarding candidates. The MAC layer (MAC) is implemented in the *ODMAC* module which implements the

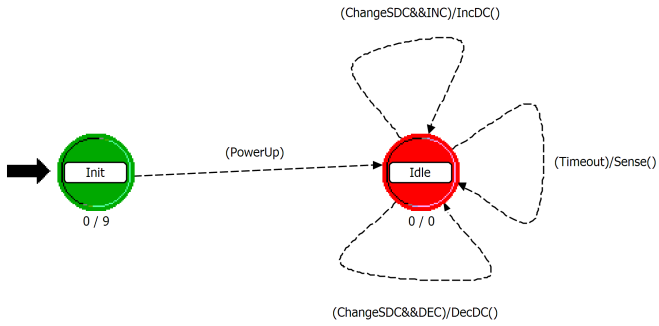


Figure 4.4: The process model of the *Sensor* module.

ODMAC protocol with the functionalities of adaptive duty cycles and opportunistic forwarding. The queue of the communication system is also implemented in this layer.

The MAC module receives and transmits packets to the PHY which consists of a radio transmitter module, a radio receiver module and an omni-directional antenna module. Finally, there is a control channel between *ODMAC* and the *Sensor* module through which the MAC protocol orders the application to increase or decrease the sensing duty cycle. There is also a control channel between the MAC protocol and the physical layer through which the process is informed about the status of the channel and uses this information for carrier sensing purposes. The *ODMAC* module also implements an outdated collision avoidance mechanism that is based on the BEB algorithm (see Section 5.2). The evaluation of the latest collision avoidance functionalities of ODMAC is presented in Chapter 5.

According to ODMAC specifications, two different packet types need to be defined, namely the beacon packet and the data packet. The beacon packet is 4-bit long and includes the layer (RAd), which denotes the distance of the node to the sink expressed in number of hops (see LAR in Section 3.6.4). The data packet consists of a 4-bit long header which contains the layer (RAd) and a 1020-bit long payload which includes data fields for the measurement and identification information for the sensor node that generated it.

4.5.1 Application Layer (APP): Sensor and Sink Process Models

Figure 4.4 depicts the implementation of the *Sensor* module as an FSM. To avoid undesired synchronizations, the first packet is scheduled randomly. Then, the packet generation is scheduled based on the attribute PGP, which is the sensing period s. Again,

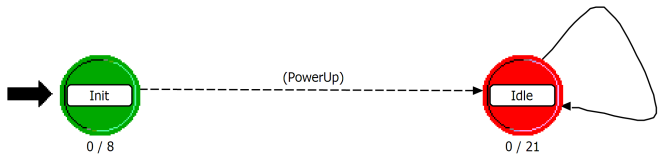


Figure 4.5: The process model of the *Sink* module.

the period is randomly increased by 0% to 10% of the PGP value in order to avoid synchronizations. Every time the timer expires, a new packet is generated and sent to the *ODMAC* module. In parallel, the *Sensor* process responds to orders originating from *ODMAC* to adjust the sensing duty cycle (i.e. the PGP). Via a state variable, the process keeps track if there is a need for an increase or a decrease on the sensing rate.

Figure 4.5 depicts the implementation of the *Sink* module as an FSM. After initialization, the process only responds to the event of receiving packets from the *ODMAC* module. Upon a packet reception, the process destroys the packet for efficient memory management and updates the statistics accordingly.

4.5.2 Link Layer (MAC): ODMAC Process Model

Figure 4.6 depicts the implementation of the *ODMAC* module as an FSM. The process model has two main functionalities, namely receiving and transmitting. Note that the sink node never enters in the states related to the transmission mode, because its queue is always empty.

Unless an event happens, *ODMAC* is always in the *Sleep* state in which the transceiver is supposed to be turned off. In this state, any packet received from the PHY layer is discarded. This way, the deactivation of the radio is simulated. Whenever the beaconing duty cycle timer expires (TODC), the process attempts to transmit a beacon. The period of this cycle is defined by the attribute *Tdc* which implements the beaconing period t. The states *RxAwake* and *RxListen* implement the CCA functionality. If the channel is occupied the process returns to the *Sleep* state. Else, it enters the *TxListen* state in which it waits for a packet reception for a predefined time (Ttx). If the timer expires without any successful packet reception from the PHY the process goes to the *Sleep* state through the *TxAwake* state. Else, the received data packet is handled. If the node is a sink, the packet is forwarded to the APP layer. If the node is a sensor, the packet is queued. After a successful packet reception, the process moves to the *RxAwake* state and immediately retransmits a new beacon. Finally, any new packet received from the APP layer while the process is in either of the aforementioned states,

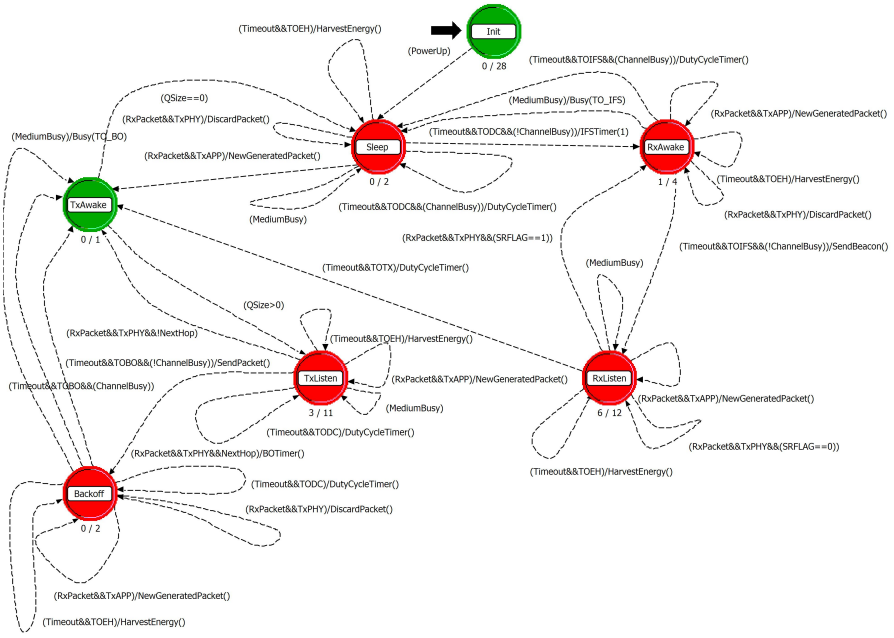


Figure 4.6: The process model of the *ODMAC* module.

it is queued for a future transmission attempt.

In the *TxAwake* state the process checks if there are queued packets that need to be transmitted. There are two ways that the process can enter this state. The first is after a reception of packet from the APP layer while being in the sleeping state. The second is right after a successful data packet reception, as described in the previous paragraph. In the *TxAwake* state, the process checks the queue size. If it is equal zero, then it goes to the sleeping state. If it is greater than zero then it enters the *TxListen* state, in which the process is waiting for a beacon from any of the forwarding candidates. Any other beacons received from the PHY layer are discarded. Right after the reception of an appropriate beacon, the process moves to the *Backoff* state, which implements collision avoidance functionalities. Then, the process transmits the packet and goes back to the *TxAwake* state. Finally, any new packet received from the APP layer while the process is in either of the aforementioned states, it is queued for a future transmission attempt.

4.5.3 Energy Model

The energy model has two aspects: energy harvesting and energy consumption. For simplicity, both functionalities are built inside the *ODMAC* module. A battery level counter is defined and every time that the process is transmitting a packet or receives a packet while being in a state that the transceiver is on, the respective amount of energy is being deducted from it. Note that the energy model also takes into account the energy consumed for the reception of discarded packets. Moreover, whenever the process tracks the amount of time it stays in a listening state in order to calculate the energy consumption in idle listening. The energy harvesting functionality is modeled as a periodic increment of the battery level counter.

The model parameters, used in the simulations, are based on a study on the energy consumption of a wireless sensor node published in [91].

4.5.4 Duty Cycle Adaptation

The adaptation of the two duty cycles is also implemented in the *ODMAC* module. The process compares the current battery level to an optimum battery level. If the current battery level is higher than the optimum, the energy consumption and thus the performance are increased and vice versa. A user-selectable variable decides whether the sensing tasks of the node are more important than the relaying tasks. If it is decided that the sensing tasks are more important, the energy consumption is increased by decreasing the sensing period (PGP) and it is decreased by increasing the beacon period (Tdc). If it is decided that the relaying tasks are more important, the energy consumption is increased by decreasing the beacon period (Tdc) and it is decreased by increasing the sensing period (PGP).

4.5.5 Node Models

Figure 4.7 depicts the node models of the sensor and sink nodes. The sensor node model consists of a *Sensor* module, an *ODMAC* module and the physical layer modules, namely a radio receiver, a radio transmitter and an antenna. A standard omnidirectional antenna from OPNET's library is used. The transmission rate and the transmission power, in the radio transmitter, is set to $1Mbps$ and $10dBm$, respectively. The sink node model is similar to the sensor node model, excluding the application layer.

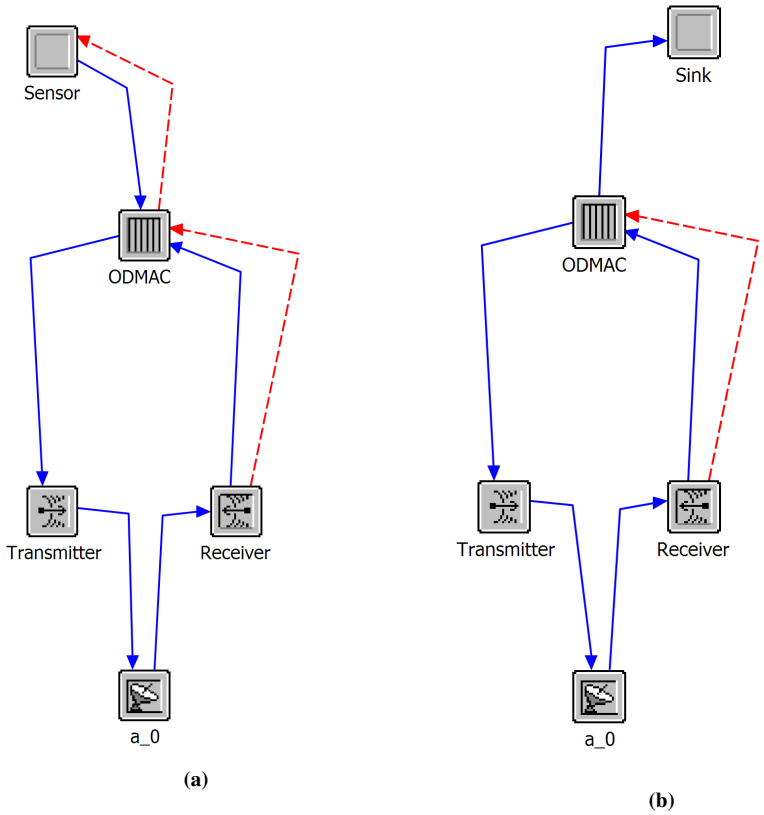


Figure 4.7: The node models of the sensor node (a) and the sink (b) node.

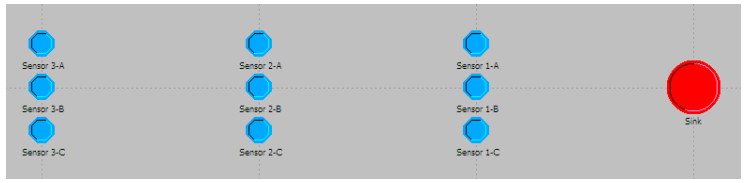


Figure 4.8: The simulated topology consists of a sink node and 9 sensor nodes positioned in 3 layers.

4.5.6 Topology: Network Model

The topology consists of one sink node and 9 sensor nodes which are placed in three layers, as shown in Figure 4.8. The distance between the nodes are placed accordingly so that the nodes of each group can only communicate directly with the nodes of neighboring group(s). Hence, the packets that are generated by the third group need to traverse three hops to reach the sink. Note also that each node has three forwarding candidates.

4.6 Evaluation through Simulations in OPNET

In this section, we present simulations that support the key findings of the analysis. In particular, we show that (i) the system is able to reach a sustainable state, (ii) ODMAC can adapt the performance of the system in various energy harvesting levels using the harvested energy for the different purposes, and (iii) Opportunistic Forwarding is able to distribute the load to the nodes that harvest more energy.

4.6.1 Achieving Sustainable Operation

First, we show that the system is able achieve a sustainable state that maximizes the performance (ENO-Max). All the sensors are set to harvest energy at a rate of $400\mu W$. Figure 4.9 shows HCR over time of one representative node of each group, as each sensor adapts its duty cycles. Note that the operating state of all the nodes gradually converges to 1, which denotes an ENO-Max state. Furthermore, we can see that the sensors that are positioned in the outer layer (represented by Sensor 3-B) converge faster to the ENO-Max state. The reason of that is that these nodes do not have forwarding tasks. As a result, their energy consumption does not depend on the sensing rate of other nodes. These sensors use the surplus of energy to increase their sensing rate. This action leads to a decrease of the energy ratio of the inner nodes (represented by Sensors 1-B and 2-B). Nevertheless, after some time the inner nodes manage to stabilize the energy consumption by adapting their tasks.

4.6.2 Power Input vs. Application Performance

The next series of simulations demonstrate how the system is able to adjust the performance to the available ambient energy, and how ODMAC can be tuned to favor specific performance metrics. In the simulations all sensors but one have static duty

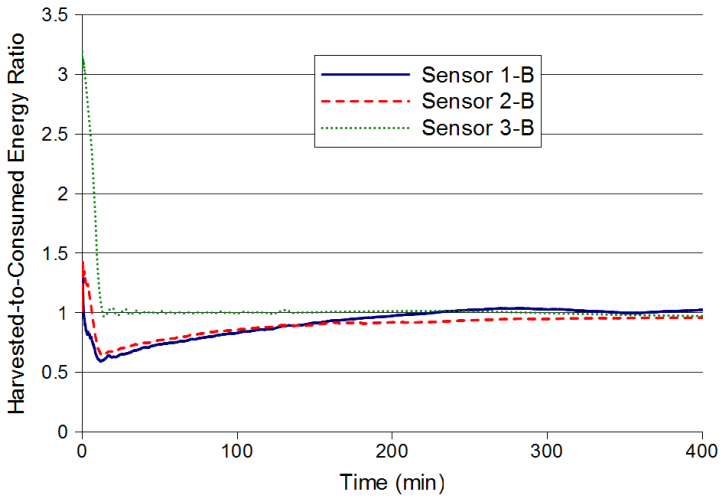


Figure 4.9: Nodes converge to a sustainable state.

cycles which are set to a sensing period of 0.6 seconds and a beaconing period of 0.2 seconds. Sensor 1-B is the only node whose dynamic duty cycle functionalities are activated. Additionally, it is exposed to different levels of energy to harvest. Each run simulates 4 hours of operation.

Figure 4.10 shows the results on the average node-to-sink delay and the average sensing period of the nodes in the outer layer for various power inputs. Generally, we notice a decreasing trend in both performance metrics, which shows that the harvested energy is effectively used to improve the performance. If the operator uses a configuration that favors shorter delays (circles), the harvested energy is used to improve the delay while the sensing period remains at high levels. On the other hand, if the operator uses a configuration that favors lower sensing periods (squares), the harvested energy is used to improve the sensing period while the delays remain at high levels.

4.6.3 Distributed Load Balancing

The last simulations aim to show how opportunistic forwarding helps nodes to distribute the relaying tasks according to their energy capabilities by simply adjusting their beaconing period. The simulation setup is similar to the previous simulation that favors shorter delay. Figure 4.11 depicts the percentage of the packets forwarded by Sensor 1-B over the total number of packets that need to be forwarded by nodes in the first layer. The other packets are relayed by the two other nodes (Sensor 1-A and 1-C).

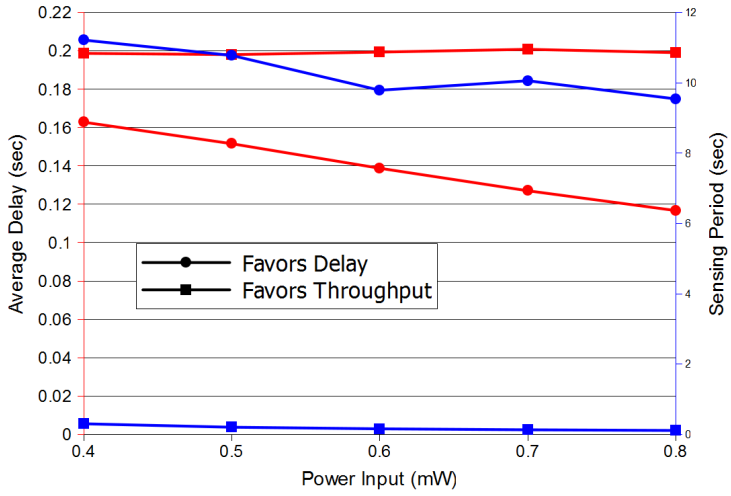


Figure 4.10: The average node-to-sink delay and the average sensing period as Sensor 1-B is exposed to different levels of power input.

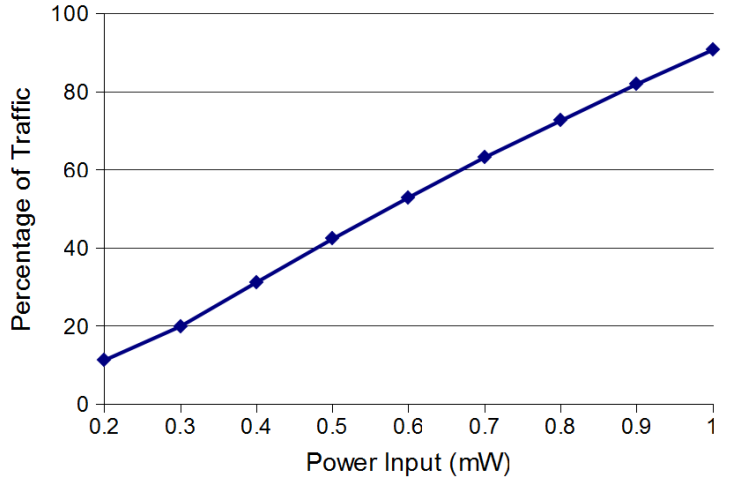


Figure 4.11: Load balancing on the forwarding duties of the sensor nodes. The figure depicts the percentage of forwarding duties carried out by Sensor-1B as it is exposed to different levels of power input. The other two nodes of the first layer are exposed to a constant power input of 0.4mW.

The figure demonstrates that when Sensor 1-B is exposed to less energy than its neighbors and increases the beacon period becomes less likely to forward traffic. The other

sensors, which have a higher beacon frequency, effectively relay more packets. On the other case, Sensor 1-B gradually carries out the majority of the forwarding tasks.

4.7 Evaluation Summary

In this section, we presented the evaluation of ODMAC focused on its Adaptive Duty Cycles (Section 3.2) and Opportunistic Forwarding (Section 3.3). The presented evaluation is conducted through mathematical analysis and simulations in OPNET.

The results from both sources indicate that sensor nodes are able to adapt their operation to sustainable levels in various realistic energy conditions. At the same time, any excess of energy can be used to favor different application-specific performance priorities, such as delay and throughput. With respect to Opportunistic Forwarding, the presented analysis and simulations verify that the feature significantly reduces the energy consumed in idle listening and promotes the autonomous and fully-distributed load balancing of the forwarding duties with respect to the energy harvesting capabilities of each node.

CHAPTER 5

Collision Avoidance with Altruistic Backoff (AB)

5.1 Evaluation Overview

In this chapter, we focus on the evaluation of AB, the collision avoidance mechanism of ODMAC that was initially introduced in Section 3.4. The evaluation is primarily focused on an energy-efficiency comparison between AB and the most commonly used solution for collision avoidance in wireless networks, namely RB (Section 5.2). Section 5.3 presents simulation experiments in MATLAB that evaluate the effectiveness to avoid collisions and the energy-efficiency of the two protocols. Furthermore, it evaluates the long-term fairness of AB, i.e. its ability to give all the contending nodes equal opportunities to access the shared channel. In Section 5.4, we evaluate the ability of AB to prioritize high-priority data. Lastly, Section 5.5 summarizes the results of the evaluation.

5.2 Random Backoff (RB)

Collision avoidance in wireless networks was introduced because collision detection mechanisms, traditionally used in wired networks, are impossible. Detecting a colli-

sion while it is happening is not possible in wireless networks, because the radio is not able to transmit and receive simultaneously. Collided transmissions can only be detected by the receiver after their completion. Therefore, in high throughput wireless networks with large data packets, such as IEEE 802.11 [55], collisions lead to a significant throughput degradation.

The solution to this problem was given by avoiding collisions through RB. The idea is that the protocol defines a time interval (*timeslot*) and a CW. Before transmitting, each node selects a random number, chosen uniformly between zero and $CW - 1$, and it delays the data transmission by that amount of timeslot while listening to the channel for other transmissions. If the channel is idle, data transmission follows. If the channel gets occupied by another transmission, the node freezes the timeslot counter and backs off. When the channel becomes idle again the node unfreezes the timeslot counter and the process is repeated until the counter reaches zero. At this point, the data transmission follows. As a result, unless two transmitters select the same random number, the collision is avoided.

The size of CW is associated with a performance trade-off. If its value is too small, the probability of two nodes selecting the same random number gets high. On the other hand, if its value is too high, the transmitters waste a lot of time in idle listening, leading to protocol overhead and throughput degradation. IEEE 802.11 DCF [55] solves this problem by adapting CW to the level of contention. This mechanism works as follows. CW is initialized with a small value, which is doubled every time a collision occurs (with a maximum limit) and gets back to its minimum value after a successful transmission. This mechanism is called BEB and results to a low CW in low contention that can quickly increase in the case of traffic bursts.

Receiver-Initiated MAC protocols for WSNs inherited the principle of RB from traditional wireless protocols. RI-MAC [107] and many other receiver-initiated MAC protocols (see Chapter 2) adopt variations of RB for collision avoidance. Given the establishment of RB as the state of the art solution for collision avoidance, the evaluation of AB will be mainly focused on a comparison study to it.

5.3 Evaluation of Energy-Efficiency and Fairness

In this section, we evaluate the proposed collision avoidance mechanism, AB, by comparing it with RB. The key difference between the two mechanisms lies in the way the collision is detected. Having energy efficiency as our metric of interest, we focus the comparison on how much time the nodes spend on idle listening. In the case of AB, idle listening is the time a sender waits for a beacon. In the case of RB, idle listening is the time a sender waits for a beacon plus the number of timeslots it waits afterwards.

We consider two variations of RB, namely Constant Backoff (CB) and BEB. In CB, the CW is fixed to a constant value (cw). In BEB, CW follows the binary exponential approach and cw represents the minimum contention window (CW_{min}).

5.3.1 Simulation Setup

We model and simulate the two methods as follows. We consider one single receiver that transmits beacons at a set frequency and a set of n nodes that are using these beacons to send their data. A round consists of the time between two beacon transmissions. Every round, each node has a probability to generate data that is equal to the ratio of the beaconing period of the receiver over its sensing period. Nodes with data wake up at a random time during the round and the time up to the following beacon or ABR reception is considered idle listening. In the case of AB, a collision happens when two nodes transmit the ABR at the same time frame. In the case of RB, a collision happens when two or more senders select the same and lowest random number. We set the duration of the timeslot at $100\mu s$ and the maximum CW_{max} at 64. Unless stated otherwise, we assume that, upon a backoff event, nodes buffer the packet and attempt to retransmit it together with the next generated packet. The simulations are conducted in MATLAB.

5.3.2 Collision Avoidance Efficiency

At the beginning, we fix the beaconing period of the receiver (BP) to 4 seconds and the transmission attempt period of the receivers (SP) to 20 seconds. Figure 5.1 shows the collision rate of the different schemes (calculated after 10000 rounds). BEB is preventing more collisions than CB for low contention windows (cw) but the difference decreases as the cw increases. This happens because as the cw increase, the probability of two or more nodes selecting the same random number decreases and, as a result, the need to double the contention window decreases. Decreasing the number of contending nodes has a similar effect. When the contention is low, a constant contention window performs sufficiently well.

AB appears more able to avoid collisions. This happens because of the random channel access. In other words, a collision can happen only if two or more nodes send an ABR at the same time. Due to the fact that the ABR transmissions are randomly distributed in time, a simultaneous ABR transmission is less probable than selecting the same random number in RB. For the same reason, increasing the contention window brings the performance of BEB and CB closer the performance of AB.

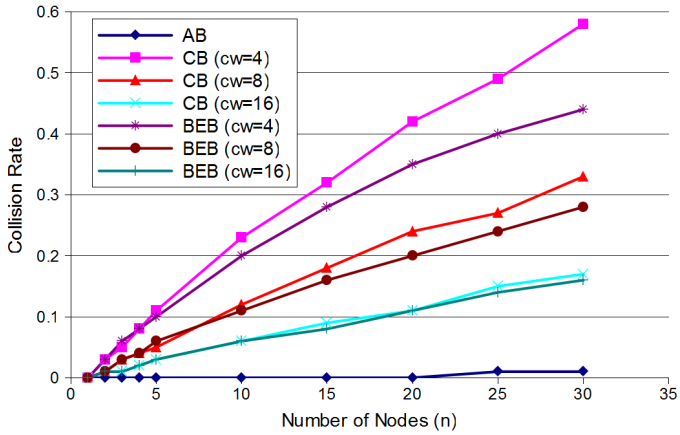


Figure 5.1: Collision rate of Altruistic Backoff (AB) and Random Backoff with constant (CB) or binary exponential (BEB) contention window. In the case of CB, cw represents the constant contention window. In the case of BEB, cw represents the minimum contention window.

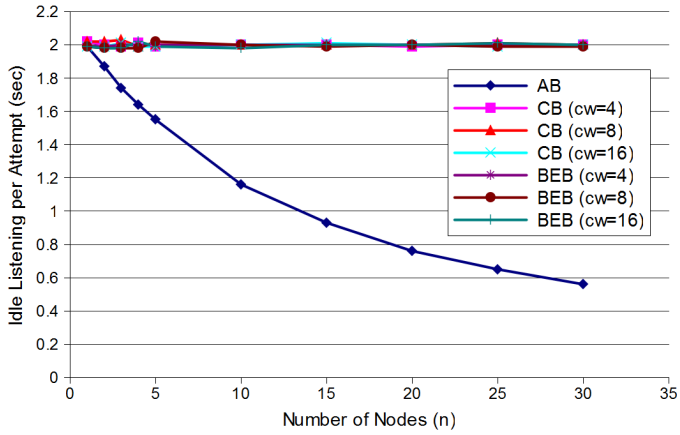


Figure 5.2: Average idle listening per transmission attempt of Altruistic Backoff (AB) and Random Backoff with constant (CB) or binary exponential (BEB) contention window. In the case of CB, cw represents the constant contention window. In the case of BEB, cw represents the minimum contention window.

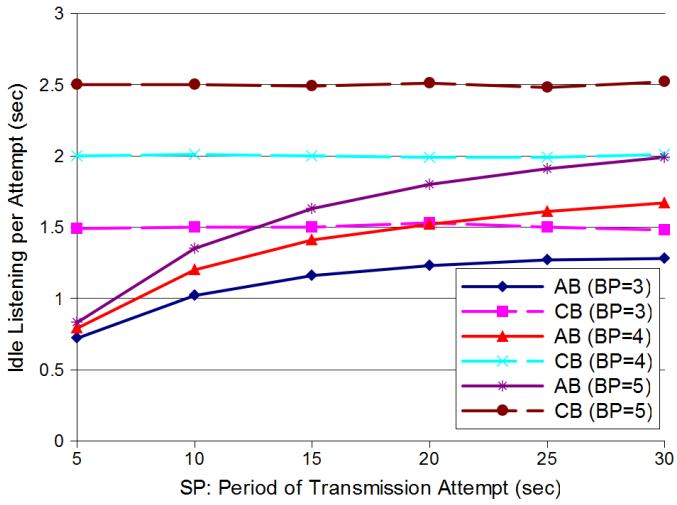


Figure 5.3: Average Idle Listening per transmission attempt for Altruistic Backoff (AB) and Random Backoff with constant (CB) contention window. BP represents the beaconing period of the receiver in seconds. SP represents the period of a transmission attempt.

5.3.3 Idle Listening Mitigation

Figure 5.2 shows the average idle listening per transmission attempt on the same simulation. Notice that CB and BEB show a constant behavior that does not increase with neither the number of nodes nor with the contention window. The average idle listening is equal to half the period of beaconing ($BP/2$). Intuitively, we expect the idle listening to increase as the contention window increases, due to the contribution of the additional timeslots. However, the results indicate that the impact of increasing the contention window is insignificant. This behavior is explained by the size of the timeslot ($100\mu s$) with respect to the expecting time a sender waits for a beacon. In other words, the contribution of the initial idle listening for the connection establishment is orders of magnitude higher than the contribution of any additional timeslots.

The figure shows that, in the case of AB, the average time the sender spends in idle listening decreases as the number of nodes increases. The more contention, the more ABR frames are transmitted and the faster contending nodes back off. Notice that the average idle listening for AB is half the period of beaconing ($BP/2$) when there is no contention ($n = 1$).

The above results indicate that it is sufficient to consider only one version RB to study idle listening. In Figure 5.3, we consider 5 contending senders and CB with fixed

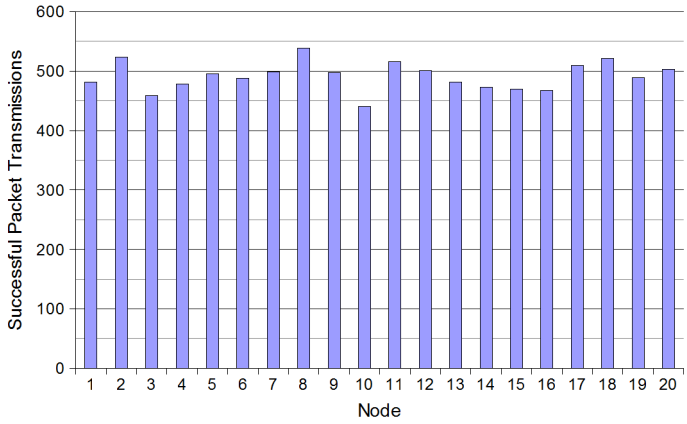


Figure 5.4: The distribution of successful transmissions indicates that AB provides long-term fairness.

contention window ($cw = 4$). We variate the period of a transmission attempt (SP) and the period of beaconing (BP). The results show a similar constant behavior for CB, while the average idle listening of AB decreases as the traffic increases (SP decreases).

5.3.4 Validation of Fairness

Figure 5.4 shows the distribution of successful transmissions over all the contending nodes, considering $n = 20$, $BP = 4s$ and $SP = 20s$, for the case of AB. We can observe that random channel access leads to equal probabilities for every node to be the last sender to wake up before the beacon. Therefore, AB provides long-term fairness for channel access.

5.4 Evaluation of Traffic Differentiation

In this section we evaluate the ability of AB to differentiate traffic to provide QoS. We consider two classes of traffic, namely *High Priority* and *Best Effort*. Sensor nodes mark the data packets that they generate with either one of the two priority classes.

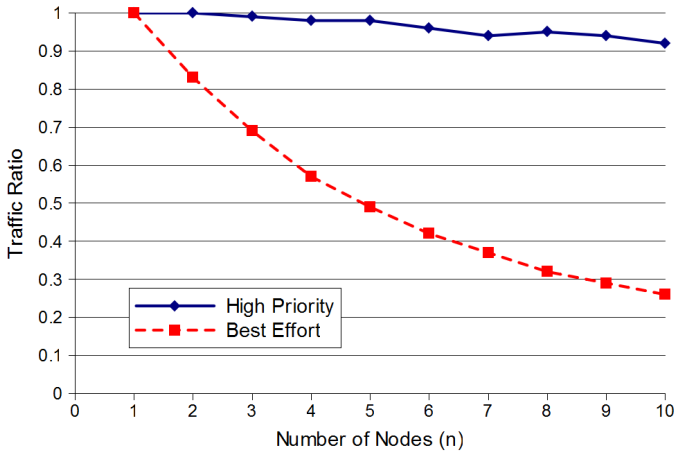


Figure 5.5: The average ratio of the amount of data packets that take a beacon over the total amount of generated packets for each priority class. As the contention increases, the protocol sacrifices *Best Effort* traffic for *High Priority* traffic.

5.4.1 Simulation Setup

We model traffic differentiation with AB similarly to Section 5.3. We consider one single receiver that transmits beacons at a set frequency and a set of n nodes that are using these beacons to send their data. A round consists of the time between two beacon transmissions. Every round, each node has a probability to generate data that is equal to the ratio of the beaconing period of the receiver over its sensing period. Nodes with data wake up at a random time during the round. Moreover, nodes mark the data that they generate as *High Priority* with a probability p. According to the protocol specification (see Section 3.4), the sensor node that wakes up last and has a data packet marked as *High Priority* takes the beacon. If there is no sensor node with *High Priority* data packets contending for the medium, the sensor node that wakes up last and has a data packet marked as *Best Effort* takes the beacon.

5.4.2 Priority of Urgent Traffic

For the following simulation, we consider $p = 0.05$, $BP = 1s$ and $SP = 3s$. Figure 5.5 shows the average ratio of the amount of data packets that take a beacon over the total amount of generated packets, for each priority class (calculated after 10000 rounds). As the contention increases, more *Best Effort* traffic backs off, while the *High Priority* traffic is less affected. Essentially, AB sacrifices less important traffic to pri-

oritize urgent traffic. The slight decreasing trend, in the case of *High Priority* traffic, is attributed to the rounds that multiple nodes with *High Priority* traffic contend with each other.

5.5 Evaluation Summary

The results indicate that AB is long-term fair and scales well with increasing levels of contention, as the ABR frames efficiently put the contending nodes to sleep early and less energy is wasted in idle listening. Collisions are less likely to happen, as AB effectively restores the beneficial aspects of random channel access in collision detection and avoidance. Furthermore, AB is able to prioritize urgent traffic, such as alerts, by sacrificing less important data packets.

CHAPTER 6

Security Extensions: Receiver Authentication Protocol (RAP)

6.1 Evaluation Overview

In this section we evaluate RAP that is introduced in Section 3.5. In particular, Section 6.2 presents the beacon replay attacks and motivates why standard countermeasures for replay attacks are not applicable for the case of ODMAC and other receiver-initiated MAC protocols. Section 6.3 documents the formal verification of RAP using two protocol verification tools. Section 6.4 models the resilience of RAP in space exhaustion, as well as the energy consumption overhead of RAP's modes of operation. Lastly, Section 6.5 summarizes the evaluation.

6.2 Motivation and Related Work

A replay attack is defined as an attack against a protocol where previously exchanged messages are reused in order to fool legitimate participants into thinking that the current

run of the protocol is valid and exchanged data is fresh [24]. The replay attack is a well known threat for WSNs. It can be used as a building block for other attacks such as Path DoS [23] where a whole path from one sensor node to the base station is filled with bogus packets. Furthermore, depending on the specific application that is being run on top of the network, replayed data messages could pose different kind of threats according to their specific meaning. One of the well known security suites for WSNs, TinySec [60], explicitly leaves replay attacks out of consideration.

Replay attacks can be deployed against ODMAC or any other receiver-initiated MAC protocol. The key idea is to capture and replay beacon frames. Among other things, beacons contain the identity of their creator which is the main piece of information needed to determine whether or not a specific beacon can be used by a potential sender, according to the overlying routing algorithm. By replaying beacons, it is possible to deploy a series of other attacks.

First of all, it is possible to flood the channel with these frames, trying to accumulate as many data packets as possible, therefore performing what is known as a *Sinkhole attack* [59]. After the acquisition, packets can be completely dropped thus performing a *Blackhole attack* [59]. A subtler possibility is to implement a *Selective Forwarding attack* [59] (sometimes also called *Grayhole attack*), where the packets are not dropped indiscriminately, but rather according to their source. This yields a harder to detect and yet still very effective attack.

Another possible attack is the *Sybil attack* [59], where a node relates to other nodes with more than one identity. This could lead to routing paths to be invalidated, or even nodes that are physically not within range one another, to be led to believe so; turning this into a rudimentary one-man *Wormhole attack* [59].

One last meta-attack, specific to duty-cycling wireless networks, is what we call the *Sleepwalker attack*. The idea behind this attack is that any of the aforementioned attacks can be deployed by a malicious node that is within range of the attacked node, without being detected by the latter. Beacons can be collected from a receiver node and replayed in the same neighborhood when the original node is asleep.

Other previous works have addressed and mitigated replay attacks. The most common solution is to make each packet unique by means of adding either a counter or a timestamp. Timestamps are usually harder to implement because they require an agreement between the sender and the receiver which, in turns, translates to a global agreement for forwarded packets. The alternative is represented by monotonically increasing counters that are generally included within a message authentication code, making sure that each message will be different from the previous one.

The authors in [94] use both techniques, one for each part of the protocol. In the first, a counter is added within the message authentication code, whereas time synchronization

and hash chains are used in the second. Similarly, the authors in [75] use a sequence number in the message exchange. The work found in [28] makes use of hash chains and a two-step scheme, namely detection and response. For the detection part each node adds its own ID value to the message, along with an always increasing common hop count. The authors in [45] use the LEACH [49] protocol in a query driven paradigm and build upon it a mechanism that exploits the cluster organization, relaying on the cluster heads to compare timings of the messages from the registered nodes. Finally, [104] presents a time synchronization scheme that makes use a sequence number in order to prevent replay attacks.

The standard techniques to prevent replay attacks are inapplicable for beacon replay attacks in receiver-initiated MAC protocols. One of the main advantages of the receiver-initiated communication paradigm is the fact that no synchronization is needed for the protocol to operate. Timestamps, in order to be meaningful, require some form of clock synchronization among the nodes. This usually comes for free within protocols that use synchronized duty-cycles, but is a costly feature to obtain in receiver-initiated protocols.

The other common alternative is the use of counters or session numbers. The latter are random non-reusable numbers that uniquely identify a particular message, or in this case a beacon. In order to check if a received beacon is fresh or replayed, a table of all the previously used session numbers should be kept. Given the highly constrained resources of a node, and the fact that not all the beacons are received, this solution is inapplicable. Counters, on the other hand, eliminate the need of having to store a whole table, as only the latest value is needed. Upon receiving a message the new counter value can be compared against the last one received and if newer, the beacon is accepted. This mechanism is inapplicable for ODMAC because there is no way for a sleeping node to know how many beacons were sent between the current and the previous active period, allowing the attacker to replay beacons that were not received by sleeping nodes.

Message authentication codes can be used for beacon authentication, but they cannot prevent a replay attack. All that can be guaranteed upon receiving a beacon whose message authentication code correctly matches, is that the at some moment in time that beacon was genuine, created by a legitimate node and intended for another legitimate node. However, it is not possible to establish whether or not the beacon has been replayed.

All these reasons motivated *RAP*, a novel authentication scheme specifically designed to detect and prevent the beacon replay attack in receiver-initiated MAC protocols.

6.3 Formal Protocol Verification

The formal protocol verifications aims to verify that RAP effectively countermeasures the beacon replay attack. The protocol is first modeled and and then fed to the verification tools.

6.3.1 Protocol Modeling for OFMC and ProVerif

In order to formally verify RAP, we modeled it using the Alice-and-Bob (AnB) language. AnB [82] is a specification language based on the popular AnB notation for security protocols. Besides providing a way to describe the protocols of interest in a compact way, AnB is also a formal language with unambiguous semantics for the honest agents, the intruder, and the goals of the protocol. The semantics of AnB are translated in the Automated Validation of Internet Security Protocols and Applications (AVISPA) Intermediate Format [6]. The Intermediate Format can be directly read by several tools, such as On-the-Fly Model Checker (OFMC) [8]. We also manually translate the AnB specification to the abstraction-based tool ProVerif [10]. The main idea for using two tools lies in their complementary strengths. OFMC is effective in finding attacks, but can verify a protocol only for a bounded number sessions; on the other hand ProVerif abstracts from the concrete search space, sometimes producing false attacks (especially for replay-protection goals), requiring adaptations of the specification. Therefore, verifying the protocols with different approaches gives a higher confidence.

The core of the AnB specification is the definition of the behavior of each role of the protocol when it is played by an honest agent, namely how this agent decomposes the messages it receives (and what parts of a received message it can actually check), and how the agent composes outgoing messages based on its initial knowledge and the previously received messages. Here, all variables that do not appear in the knowledge section of the AnB specification are values that are *freshly* created by the agent who first uses them. For instance in the detection protocol RAP-D, A freshly creates the challenge C and the data $Data$.

The standard intruder model of AnB is the common Dolev-Yao intruder [27] who controls the entire communication medium, it can arbitrarily overhear, send and even intercept messages. This is clearly inspired by communication in wired networks, and for many questions this is unrealistically strong for WSNs: an intruder may not control all locations spanned by the WSN and also it may not be able to hear a message when it is blocking it (e.g. by jamming). However, verifying the protocol under such a strong intruder gives higher confidence.

Furthermore, we use authentication goals which correspond to Lowe's injective agree-

```
Protocol: Basic Auth

Types:
Agent A,B;
Function mac,sk

Knowledge:
A: A,B,mac,sk(A,B);
B: A,B,mac,sk(A,B)

Actions:
B->A: B,mac(sk(A,B),B)
A*->*B: Data

Goals:
A authenticates B on B
```

(a)

```
Protocol: RAP-D

Types:
Agent A,B;
Function sk

Knowledge:
A: A,B,sk(A,B);
B: A,B,sk(A,B)

Actions:
B->A: B
A*->*B: Data,C
B->A: {|C|}sk(A,B)

Goals:
A authenticates B on B,C
```

(b)

```
Protocol: RAP-P

Types:
Agent A,B;
Function sk

Knowledge:
A: A,B,sk(A,B);
B: A,B,sk(A,B)

Actions:
B->A: B
A->B: C
B->A: {|C|}sk(A,B)
A*->*B: Data

Goals:
A authenticates B on B,C
```

(c)

Figure 6.1: The protocols used in OFMC described with AnB notation. A basic authentication model (a) is only enough to prevent beacon forgery. RAP-D (b) and RAP-P (c) are not affected by beacon replay attacks.

ment [76]. For the concrete example of the goal $A\ authenticates\ B\ on\ B, C$, as soon as B learns the fresh challenge C, it produces (in our model) an auxiliary event $witness(B, A, C)$ formalizing the intention to run the protocol with A and using challenge C. When A successfully finishes her run of the protocol, she produces also an event $request(A, B, C)$ to formalize that she finished the protocol, apparently with B and using challenge C. It counts as an attack if a trace contains more request events than corresponding witness events, i.e., when A either believes in receiving something from B that B actually has never sent, or if A is tricked into accepting something more times than B actually sent.

Finally, we use Maurer's channel notation [80], which is supported by the AnB language (for the formal definitions in AnB see [84]). Informally $A \bullet\!\!\rightarrow B$ means that A sends a message *authentically* to B (so B can be sure it really comes from A and was meant for B), $A \rightarrow\!\!\bullet\ B$ means that the message is sent confidentially (so A can be sure only B can receive it), and $A \bullet\!\!\rightarrow\!\!\bullet\ B$ means both authentic and confidential transmission. We use this notation to abstract from how the transmission of the actual data is organized, i.e., how authentication and confidentiality is achieved if they are desired. In fact, this problem is orthogonal to the replay-protection for the beacon that we study here, and the channel notation allows us to abstract from that.

Figure 6.1 shows the models of RAP using the AnB notation [82]. It should be noted that we decided to strip down the protocols in order to focus the attention on the beacon

replay attack, hence we kept only the messages relevant in this sense. Furthermore, we omit the basic version of the protocol which does not include any form of authentication, as it yields the trivial attack of beacon forgery.

6.3.2 Protocol Verification with OFMC and ProVerif

In the case of basic authentication (Fig. 6.1a), OFMC can detect the beacon replay attack, shown in Fig. 6.2, within a few seconds. For the intruder i it is simply enough to store a previously received beacon and replay it to a victim node in order to receive the data. Another interesting fact is that by adding the *weakly* clause to the authentication goal, turning it into Lowe's non-injective agreement [76], no attack is found. This helps to build confidence in the model and its correctness. When running OFMC on RAP-D and RAP-P, the protocols are verified for 3 sessions in 2 and 24 minutes respectively, without any attack. Note that in each session, OFMC considers all possible instantiations of the roles with concrete agents, both honest and the intruder. Thus, whenever a protocol is verified for a given number of sessions, then there is no instantiation of the roles for these parallel sessions that can lead to an attack. As a rule of thumb, attacks are usually detected within 2 sessions.

ProVerif computes on first-order Horn clauses [50] that represent an overapproximation of the reachable events and messages the intruder can ever learn. There is therefore no notion of timeline, posing some difficulties for the analysis of replay, even though ProVerif offers the notion of *injective* events for this purpose. We used the Application Integration Framework (AIF) framework [83] that is built on top of ProVerif and allows to specify a state-transition system with a number of sets of data. In this particular case, we define for each agent the set of challenges that are sent out and have not been responded to, as well as those that have been responded to (and are therefore used). ProVerif verifies RAP-D and RAP-P in 5 and 3 minutes respectively.

6.4 Energy Consumption Analysis

After verifying the effectiveness of RAP, the next step is to evaluate its energy-efficiency. The analysis exposes a trade-off between energy-efficiency and the level of security of the protocol in terms of resilience to space exhaustion.

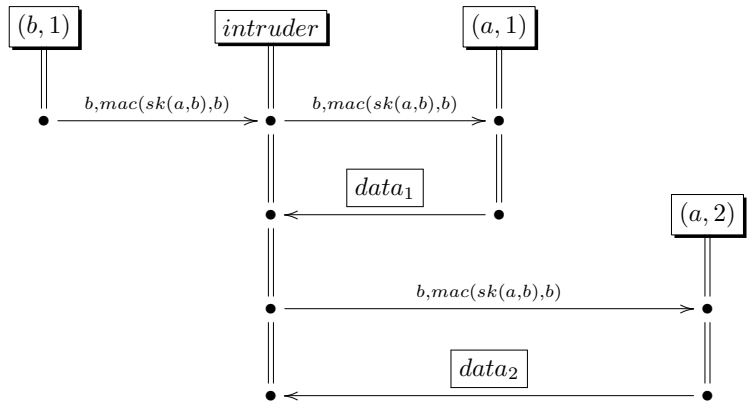


Figure 6.2: Trace of the beacon replay attack found by OFMC in the basic version of ODMAC.

6.4.1 Space Exhaustion Analysis

In this section we model and discuss the resilience of RAP to space exhaustion. An attacker can passively monitor the communication of legitimate nodes and collect pairs of challenge and response messages. This way, the attacker can gradually build a dictionary that can be used to bypass RAP. The size of such a dictionary is a direct indication of the resilience of the protocol against space exhaustion.

When RAP is in prevention mode, an attacker can trivially map the challenge to the respective response, as they are both distinct messages. Thus, the size of each word $D_{\text{RAP-P}}$ in the dictionary is equal to the size C_P of the challenge in bits, translating to $2^{D_{\text{RAP-P}}}$ words.

$$D_{\text{RAP-P}} = C_P \tag{6.1}$$

When RAP is in detection mode, we aim at a small challenge to keep the overhead low. However, the dictionary size can be significantly increased by encrypting the challenge together with the data, using CBC encryption [105]. Essentially, CBC hides the challenge within the data, preventing the attacker from mapping the challenge to the response. As a result, a dictionary can only be built by mapping the whole message (that contains both the data and the challenge) to the respective response. Therefore, the size of each word $D_{\text{RAP-D}}$ in the dictionary, which translates to a dictionary size of

$2^{D_{\text{RAP-D}}}$ words, is equal to the aggregate size L_D of the data and C_D of the challenge.

$$D_{\text{RAP-D}} = C_D + L_D \tag{6.2}$$

As an attacker can force the system to change the mode of operation, we note that the overall resilience of RAP to space exhaustion is equal to the smallest of the two dictionaries, $D_{\text{RAP-D}}$ and $D_{\text{RAP-P}}$. Furthermore, the sizes of the two challenges, C_D and C_P, which constitute configurable protocol parameters, define the level of security in the same manner the size of a key defines the level of security of an encryption algorithm. In the following section, we attempt to model the energy overhead of RAP and highlight the trade-off between security and energy-efficiency.

6.4.2 Energy Consumption Overhead Analysis

Let L_D be the size of a data packet in bits, L_B be the size of a beacon in bits and R the transmission rate of the radio in bits per second. Additionally, let P_{tx} and P_{rx} be power consumption for transmitting and receiving / listening respectively. First, we estimate the energy consumption for a single packet transmission in the case of not using RAP. For the receiver, B, the energy consumption is estimated by (6.3), where t_G is a time guard during which the radio is turned on while waiting for a answer right after a transmission. The purpose of such a guard is to account for the propagation and the processing delay.

$$E_B^{\text{Default}} = \frac{L_B}{R} P_{tx} + t_G P_{rx} + \frac{L_D}{R} P_{rx} + \frac{L_B}{R} P_{tx} \tag{6.3}$$

For the sender, A, the energy consumption is estimated similarly.

$$E_A^{\text{Default}} = \frac{L_B}{R} P_{rx} + \frac{L_D}{R} P_{tx} + t_G P_{rx} + \frac{L_B}{R} P_{rx} \tag{6.4}$$

Note that this energy model disregards the energy consumed while the sender awaits for the beacon, as this source of energy consumption is independent of the security protocol.

In the case of RAP-D, the energy consumption for a single packet transmission, for the

receiver (B) and the sender (A), is given by the following formulae.

$$E_B^{\text{RAP-D}} = \frac{L_B}{R} P_{tx} + t_G P_{rx} + \frac{L_D + C_D}{R} P_{rx} + \frac{L_B + C_D}{R} P_{tx} \tag{6.5}$$

$$E_A^{\text{RAP-D}} = \frac{L_B}{R} P_{rx} + \frac{L_D + C_D}{R} P_{tx} + t_G P_{rx} + \frac{L_B + C_D}{R} P_{rx} \tag{6.6}$$

In the case of RAP-P, the energy consumption for a single packet transmission, for the receiver (B) and the sender (A), is estimated similarly.

$$E_B^{\text{RAP-P}} = \frac{L_B}{R} P_{tx} + t_G P_{rx} + \frac{C_D}{R} P_{rx} + \frac{C_D}{R} P_{tx} + t_G P_{rx} + \frac{L_D}{R} P_{rx} + \frac{L_B}{R} P_{tx} \tag{6.7}$$

$$E_A^{\text{RAP-P}} = \frac{L_B}{R} P_{rx} + \frac{C_D}{R} P_{tx} + t_G P_{rx} + \frac{C_D}{R} P_{rx} + \frac{L_D}{R} P_{tx} + t_G P_{rx} + \frac{L_B}{R} P_{rx} \tag{6.8}$$

We define the Energy Consumption Overhead (ECO) of a protocol as the ratio of the energy consumption for a single packet transmission (while using the respective protocol) over the case of a plain communication (without using it). The subscript j is equivalent to B for the receiver and A for the sender.

$$\text{ECO}_j^{\text{RAP-D}} = \frac{E_j^{\text{RAP-D}}}{E_j^{\text{Default}}}\,, \qquad \text{ECO}_j^{\text{RAP-P}} = \frac{E_j^{\text{RAP-P}}}{E_j^{\text{Default}}} \tag{6.9}$$

6.4.3 Numerical Results

For the following numerical results, we assume using the CC2500 radio [116] which has the following characteristics: $R = 500\ Kbps$, $P_{tx} = 53.8\ mW$, $P_{rx} = 42.5\ mW$. Additionally, we consider the following values for the protocol parameters: $L_B = 2\ bytes$, $L_D = 32\ bytes$ and $t_G = 10\ \mu s$. Figure 6.3 shows the cost for a single packet transmission of the two protocols, as defined in (6.9). Notice that the cost of the sender and the receiver increase linearly with the challenge size while the cost for the latter is relatively higher. The difference between them also increases as the challenge size increases.

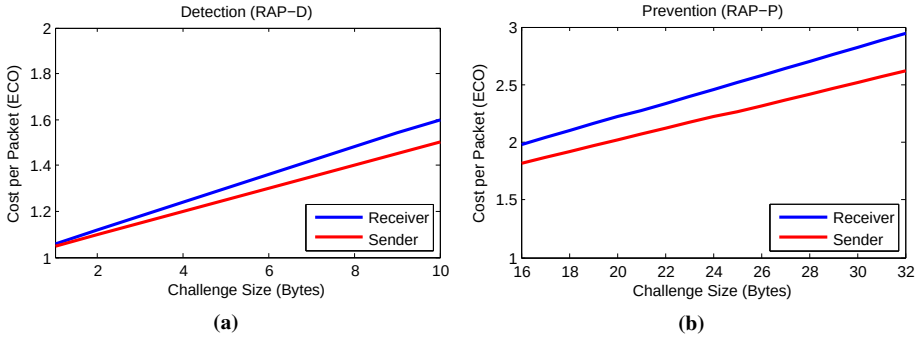


Figure 6.3: Energy Consumption Overhead (ECO) for a single packet transmission for RAP-D (a) and RAP-P (b).

In Figure 6.4, we compare the cost of RAP-D and RAP-P, showing the low-overhead nature of the former. Particularly, we compare the cost overhead ECO_B for the receiver of the two protocols keeping the same dictionary word size D, as defined in (6.1) and (6.2). Note that the dictionary word size indicates the resilience of each protocol to space exhaustion. In the case of RAP-D, we make sure the value of the challenge is at least 1 byte by setting it to $C_D = max(D_{\text{RAP-D}} - L_D, 1)$. As shown in the figure, the cost of using RAP-P is significantly higher than the cost of using RAP-D for the same level of security.

Figure 6.5 investigates the relative cost of the two protocols for different data sizes, by comparing the cost overhead ECO_B for the receiver of the two protocols. Additionally, we consider different dictionary word sizes as requirements for resilience to space exhaustion. The results suggest that increasing the data packet drops the energy cost for both protocols. The energy overhead of RAP-D can be kept at a minimal level as long as the data size is above the dictionary word size requirement.

6.5 Evaluation Summary

In this chapter, we presented the motivation behind the RAP (see Section 3.5) and we evaluated its performance. RAP is a challenge-response scheme with two modes of operation. RAP-D is a low-overhead protocol that is able to detect intruders who replay beacons. RAP-P, on the other hand, is a more expensive prevention mechanism. We validated the effectiveness of both RAP-D and RAP-P against beacon replay attacks using various verification tools. Furthermore, we have modeled the energy consumption of both protocols and we have exposed the trade-off between the level of security,

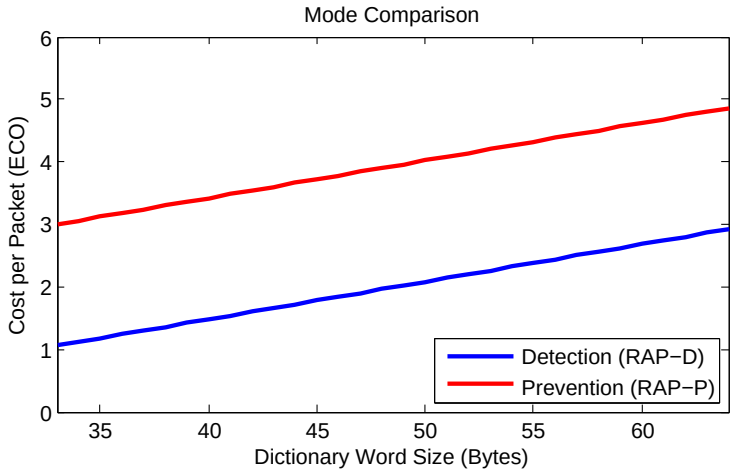


Figure 6.4: The relative cost between RAP-D and RAP-P for the same level of resilience to space exhaustion.

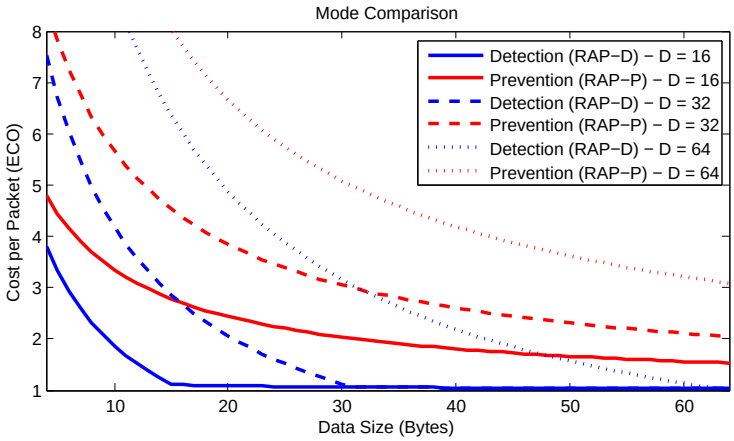


Figure 6.5: The relative cost between RAP-D and RAP-P for different data sizes (L_D) and required levels of resilience to space exhaustion (D).

measured by the resilience of the scheme to space exhaustion and the level of energy consumption. Lastly, we have highlighted the energy-efficiency nature of RAP-D, by comparing the energy consumption of the two modes of operation.

CHAPTER 7

Analytical Comparison Studies

7.1 Evaluation Overview

In this chapter, we evaluate ODMAC by conducting two analytical studies that compare its performance with state-of-the-art MAC protocols that are widely used in either the academic or the industrial world. The first study (Section 7.2) compares ODMAC with an adaptive variation of the widely-used sender-initiated X-MAC [13]. The second study (Section 7.3) focuses on an industrial application and compares ODMAC with the protocol that is currently running in a deployed industrial infrastructure used for commercial purposes. Section 7.4 summarizes the comparison.

7.2 Comparison with the Sender-Initiated Paradigm

As we discussed in Section 3.1, both asynchronous paradigms of communication (Section 1.4.2) can effectively support individually adaptable duty cycles, which is a vital requirement of EH-WSNs in order to adapt the available energy. The focus of the work presented in this section is to evaluate the receiver-initiated character of ODMAC and

identify in which conditions the receiver-initiated approach is more suitable than the sender-initiated.

In this comparison, the sender-initiated MAC protocols are represented by a basic version of X-MAC [13]. In X-MAC, the communication link is established using multiple short preambles. Instead of a long preamble, the sender is transmitting multiple short preambles that contain addressing information. The receiver is given enough time to interrupt the series of short preambles with a special packet named *pre-ack* that indicates that it is ready to receive the data. The data exchange follows. Extending the sender-initiated paradigm, X-MAC decreases the energy consumption overhead consumed for the link establishment, as the sender alternates between active and sleeping states and the receiver is allowed to interrupt the preamble transmission.

In addition to X-MAC being established as one of the most popular MAC protocols for WSNs, we choose to X-MAC to represent the sender-initiated protocols, because, contrary to other preamble protocols (e.g. B-MAC [95]), X-MAC is compatible with the mechanisms of Opportunistic Forwarding (Section 3.3). For the purposes of this comparison we consider a variation of X-MAC that incorporates opportunistic forwarding. Furthermore, we assume that neither ODMAC nor X-MAC use any active mechanisms for collision avoidance.

The analytical comparison is based on the model presented in Section 4.3. While the model effectively estimates the delays and traffic rates of both protocols, it needs to be extended with a power consumption model for X-MAC. Furthermore, we introduce the channel utilization overhead. It refers to the percentage of time a node transmits overhead data, i.e. beacons for ODMAC and short preambles and *pre-acks* for X-MAC. The channel utilization overhead indirectly approximates the amount of interference each protocol is responsible for.

7.2.1 Power Consumption Model for X-MAC

The long-term average power consumed for transmitting packets (P_i^{ttx}) of X-MAC has no difference from ODMAC and thus it is given by (4.12).

In X-MAC, the expected synchronization delay (y_i) for the sender is shared between transmitting preambles and listening for *pre-acks*, that we assume have equal size to the beacon (L_b). Thus, the long-term average power consumption of this part of the connection establishment is estimated by the following formula where P_i^t is the power consumed in transmission given by (4.13), P_{r0} is the power consumed in reception, y_i

is the waiting time given by (4.5) and r_i is given by (4.11).

$$P_i^w = \frac{1}{2} P_i^t y_i r_i + \frac{1}{2} P_{r0} y_i r_i \quad \text{[W]} \tag{7.1}$$

For each packet reception, each receiver has to transmit a *pre-ack* packet before the actual packet transmission. The long-term average power consumption for the transmission of this packet is given by the following formula where r_i^f is given by (4.10), L_b is the *pre-ack* size and R is the transmission rate. Since both the purpose of a *pre-ack* and the purpose of a beacon is to contain addressing information, we consider them to have equal size.

$$P_i^{pa} = P_i^t \frac{L_b * 8}{R} r_i^f \quad \text{[W]} \tag{7.2}$$

Lastly, each node needs to periodically listen the channel for short preambles. In the worst case scenario, the receiver starts listening while the sender begins waiting for a *pre-ack*. To account for the worst case scenario, the receiver needs to listen the channel for twice the duration of a *pre-ack* transmission. Thus, the long-term average power consumption of periodic listening is given by the following formula where t_i is the cycle period, the ratio of the preamble size (L_b) over the transmission rate (R) is the time required for a preamble transmission and P_r is the power consumed while receiving.

$$P_i^l = 2 P_r \frac{1}{t_i} \frac{L_b * 8}{R} \quad \text{[W]} \tag{7.3}$$

The long-term total power consumption of a node i, while running X-MAC, is given by the sum of (4.12), (7.1), (7.2) and (7.3).

$$P_i^{X-MAC} = P_i^{ttx} + P_i^w + P_i^{pa} + P_i^l \tag{7.4}$$

7.2.2 Channel Utilization Overhead

We define as channel utilization overhead, the percentage of time a node transmits overhead data, i.e. beacons for ODMAC and short preambles and *pre-acks* for X-

MAC. In ODMAC, the channel utilization overhead is approximated by the following formula.

$$I_i = \frac{1}{t_i}\frac{L_b * 8}{R} \tag{7.5}$$

In X-MAC, channel utilization overhead is approximated by the following formula.

$$I_i = \frac{1}{2} y_i r_i + \frac{L_b * 8}{R} r_i^f \tag{7.6}$$

The channel utilization overhead does not necessarily translates to performance degradation due to collisions. Nevertheless, the higher this metric is, the more probable is for a node to find the channel occupied while attempting to transmit.

7.2.3 Analytical Comparison

For the analysis we assume the use of LAR (see Section 3.6.4). Hence, the set of forwarding candidates for each node is set to all the sensor nodes that are one hop closer to the sink node. Moreover, we select the transmission power of each node that maximizes the number of links between the nodes. In particular, we select the maximum supported transmission power and then we gradually decrease it to the point that no links are broken.

Table 7.1: Model parameters.

L	100 Bytes	G	0 dBi	P^{tx}	10 dBm
L_b	2 Bytes	e	4	η	0.157
R	256 Kbps	P^{rx}	-96 dBm	P^{t0}	15.9 mW
f	433 MHz	A	300 m	P^{r0}	22.2 mW

Table 7.1 provides the values of the parameters of the model that are used unless otherwise noted. The parameters suppose the CC1000 radio [125]. Based on these parameters, the transmission range is approximately 105 meters. Furthermore, we consider 10 random topologies of 50 nodes that operate on the same duty cycles ($s_i \equiv s$ and $t_i \equiv t$) The sink node is placed in position $(0, 0)$. Lastly, we focus only on the power consumption overhead for the connection establishment of the two schemes. Hence, we disregard the long-term average power consumed for transmitting packets (P_i^{ttx}) which is equal for both protocols.

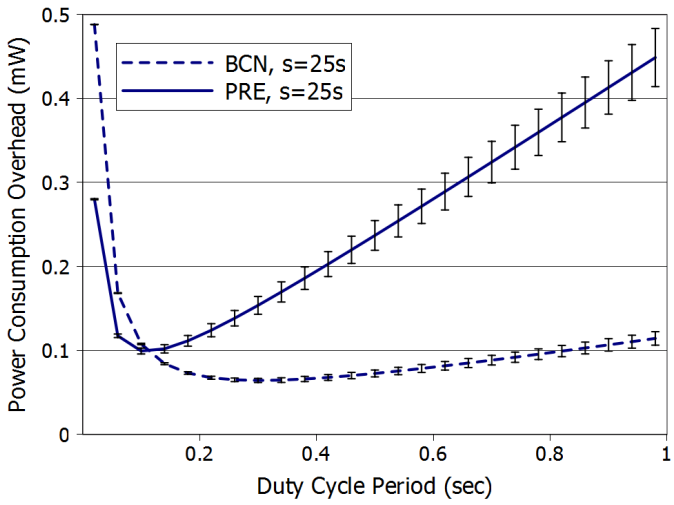


Figure 7.1: Long-term average power consumption overhead.

For the rest of the section, we will refer to ODMAC (i.e. the representative of the receiver-initiated paradigm) as the Beaconing Scheme (BCN) and to X-MAC (i.e. the representative of the sender-initiated paradigm) as the Preamble Scheme (PRE).

7.2.3.1 Basic comparison

Figure 7.1 depicts the long-term average power consumption overhead of the two protocols for different values of the duty cycle period, t, given a sensing period of $s = 25$ seconds. Generally, BCN performs better at large duty-cycling periods (t), while PRE performs better at low periods. Both schemes have a operation point where the long-term power consumption overhead is minimized. The results suggest that the beaconing scheme can be configured to consume less energy than the preamble scheme. Moreover, the minimum point of the preamble scheme appears for lower values of t, indicating shorter delays.

As a result, the beaconing scheme is more suitable in cases where either the harvested energy is relatively low or the delay is not a performance priority and the excess of harvested energy should be used elsewhere (e.g. throughput or security). On the other hand, preambles perform better in case of delay-sensitive applications in environments with high power availability.

Fig. 7.2 shows the average channel utilization overhead, which is the percentage of

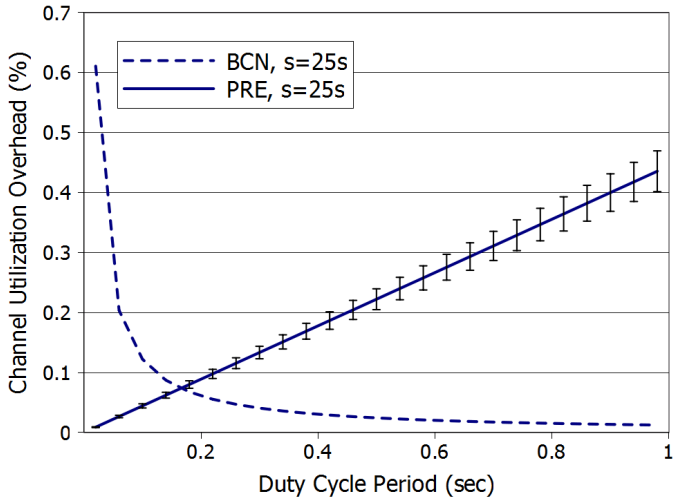


Figure 7.2: Channel utilization overhead.

time a node transmits overhead data. In low duty-cycling periods, the preamble scheme performs better due to the frequent beacon transmissions. The opposite applies for high duty cycle periods where the overhead is exponentially decreased as the beaconing period is increasing.

The error bars, in both figures, indicate the 90% confidence intervals for the average overheads over the 10 random topologies. We can observe that the beaconing scheme is more resilient to topological variations. The next section continues the analytical comparison and focuses on the influence of different values of the system parameters on the MAC schemes.

7.2.3.2 Influence of various parameters

Figure 7.3 depicts the long-term average power consumption overhead of the two protocols for different values of the sensing period (s). Decreasing the sensing period, the point of minimum consumption decreases and moves towards higher duty cycle periods for both protocols. The trends that describe their relative performance remain the same to Figure 7.1. Figure 7.4 depicts that increasing the sensing period, improves the channel utilization overhead of the preamble scheme. The result is intuitive as the main source of this overhead is the preambles themselves that depend on the amount of generated data.

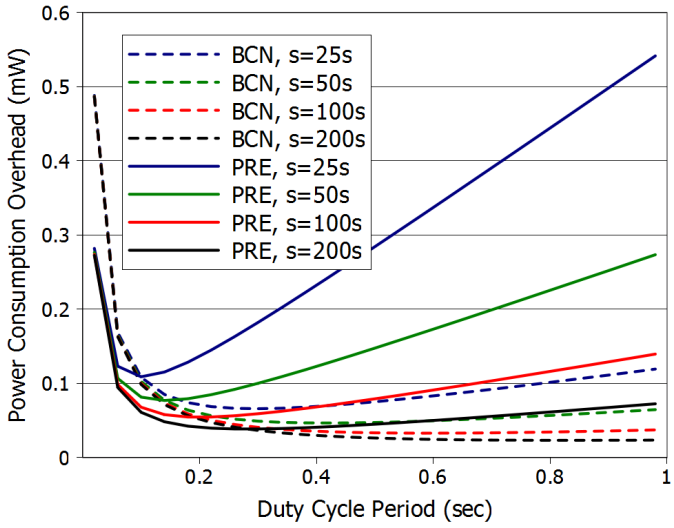


Figure 7.3: Long-term average power consumption overhead for various sensing periods (s).

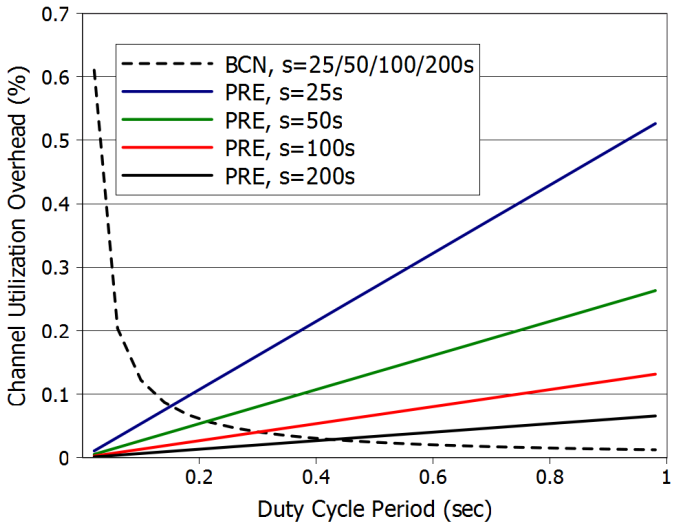


Figure 7.4: Channel utilization overhead for various sensing periods (s).

Figure 7.5 depicts the long-term average power consumption overhead of the two protocols for different values of the beacon and preamble size, respectively. Since they are both carrying addressing information, their size highly depends on the size of the

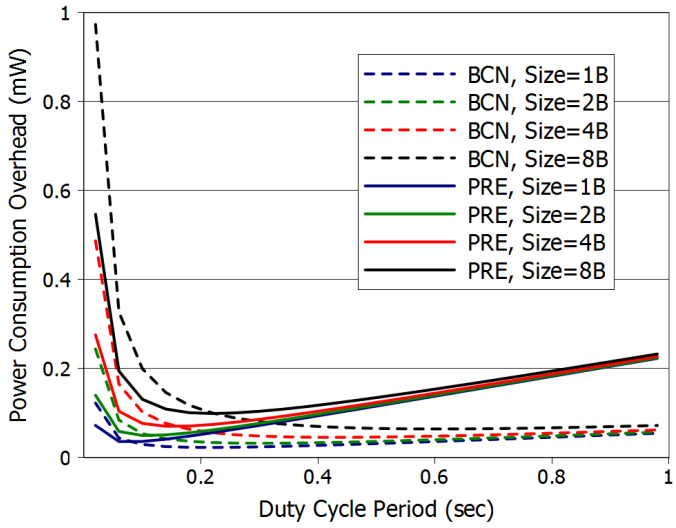


Figure 7.5: Long-term average power consumption overhead for various beacon / preamble sizes (L_b).

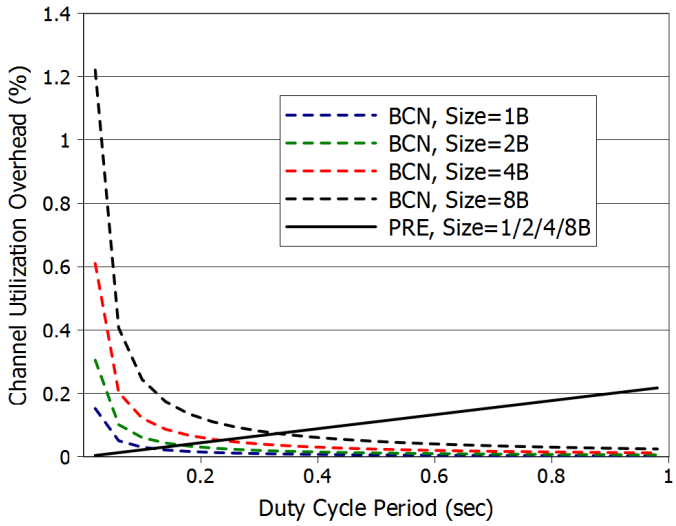


Figure 7.6: Channel utilization overhead for various beacon / preamble sizes (L_b)

network. We can observe that at the lower duty cycle periods, the smaller the beacon/preamble size the better performance of both protocols. However, smaller beacons/preambles decrease the relative difference between the MAC schemes reducing the

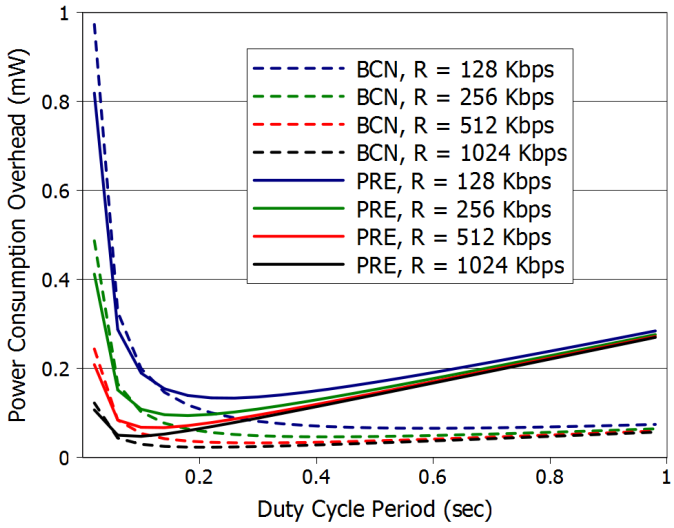


Figure 7.7: Long-term average power consumption overhead for various transmission rates (R).

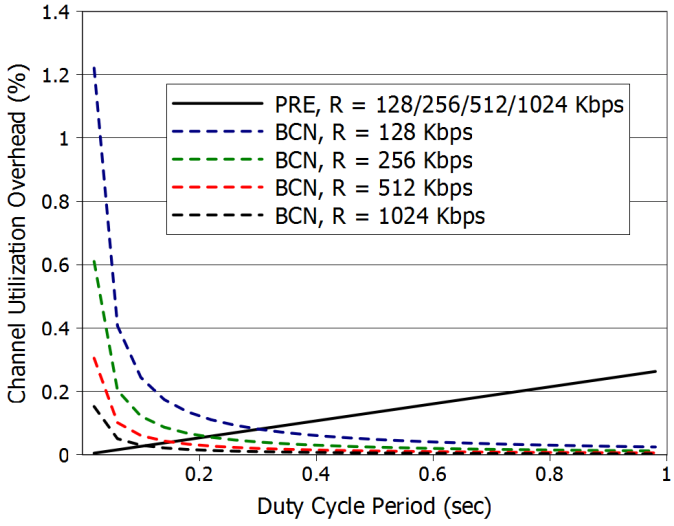


Figure 7.8: Channel utilization overhead for various transmission rates (R).

local dominance of the preamble scheme. At higher duty cycle periods the influence of the beacon / preamble size is less significant. Same conclusion applies to the channel utilization overhead (Figure 7.6).

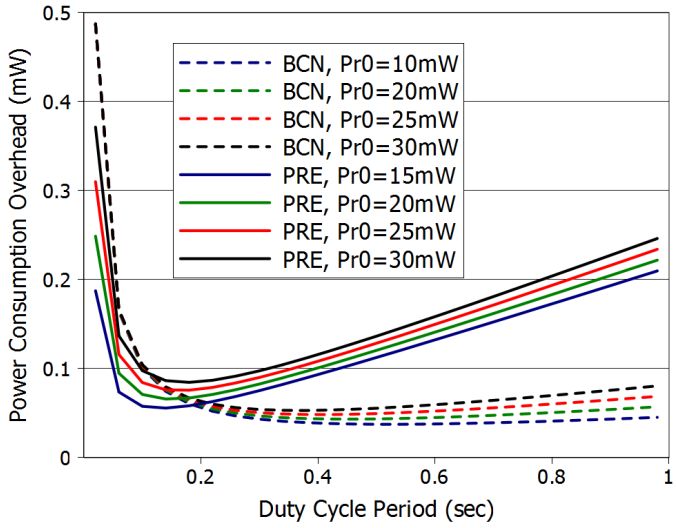


Figure 7.9: Long-term average power consumption overhead for various receiving power costs (P^{r0}).

The influence of the transmission rate on both overheads overheads (Figure 7.7 and Figure 7.8) follows a similar trend to the beacon / preamble size. In particular, as we increase the transmission rate, the power consumption of both protocols is improved. Furthermore, the improvement for the beaconing scheme is higher than the preamble scheme.

In Figure 7.9, we evaluate the long-term power consumption overhead for different values of the receiving power costs. We observe that the influence of the receiving power costs is similar for both schemes at high duty-cycling periods. On the other hand, when the duty cycle period is low, higher listening costs increase the power consumption of the preamble scheme while the beaconing scheme remains unaffected.

Lastly, we investigate the effects of the network density on the performance of the two MAC schemes. In particular, 50 to 200 nodes are placed in the same area. Figure 7.10 depicts the long-term power consumption overhead. Network density has insignificant influence on the power consumption overhead for low duty-cycling periods. However, the overhead decreases for both protocols at higher duty-cycling periods. Moreover, the improvement for the preamble scheme is higher than the beaconing scheme. Same applies for the channel utilization overhead (Fig. 7.11). Note that, for both schemes, this improvement is caused by Opportunistic Forwarding.

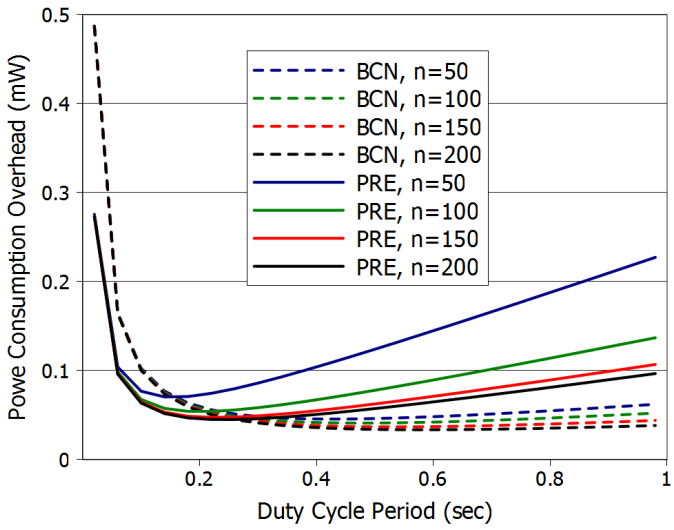


Figure 7.10: Long-term average power consumption overhead for different network densities (n).

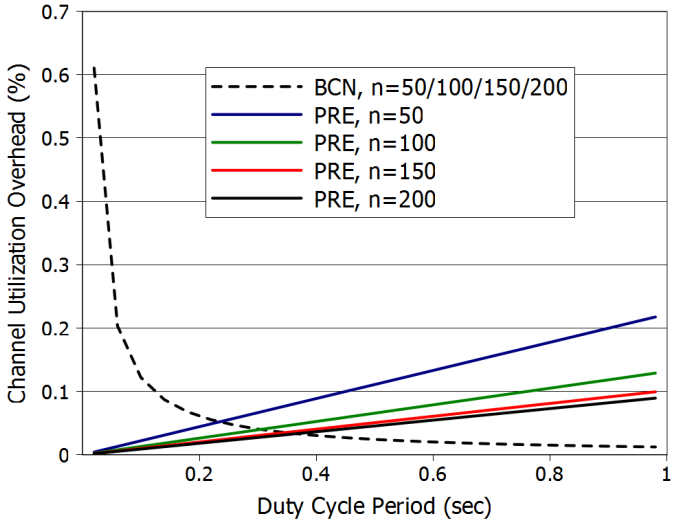


Figure 7.11: Channel utilization overhead for different network densities (n).

7.3 Industrial Case Study: Comparison with IMR+

We move our focus to Automatic Meter Reading (AMR), a commercial application of WSNs. AMR consists of embedded devices which perform time domain measurements and provide data over a unidirectional remote connection from a customer [126]. Industry is gradually moving towards Advanced Metering Infrastructures (AMIs), which refers to the entire measurement and collection system, providing bi-directional communication between the service provider and customer. The data from these systems can be used for billing purposes or as feedback into home automation systems [114] for intelligent regulation of energy sources. In this context, we are interested in industrial Energy Harvesting - Advanced Metering Infrastructures (EH-AMIs).

To this purpose, we present a real world case study conducted in collaboration with a leading company in the AMR sector, namely *Brunata A/S*. Brunata is an independent Danish exporter of solutions for individual billing of costs for heating and water, with experience in the development and production of metering equipment. There are currently more than 20 million Brunata Heat Cost Allocators (HCAs) in service, that ensure costs for heating and water are billed according to metered consumption. HCAs are mounted on radiators in Denmark and in an increasing number of countries worldwide, and thus the heat produced by the radiator is an ideal source of power for the Heat Cost Allocator (HCA). The company monitors these meters and can therefore supply accurate and fair billing information according to heat and water consumed by individual offices or dwellings.

Brunata developed an Energy Harvesting - Heat Cost Allocator (EH-HCA) prototype that mounts on radiators and harvests thermal energy from them. Additionally, they conducted experiments that estimate the energy budget of an EH-HCA. These experiments aimed to study the worst case scenario. As a result, they were conducted on LST radiators, whose surface temperature is in the range $30 - 40°$C. These radiators are widely used in environments such as kindergartens and hospitals due their safety requirements. Significantly more power can be harvested by standard radiators that heat up to $50°$C. A conservative conclusion of these experiments was that an EH-HCA can harvest between 1μW to 10μW of power from LST radiators for radio communication [121].

The analytical study, presented in this section, compares ODMAC to the MAC scheme currently used by Brunata, a protocol named IMR+, assuming the power budget of EH-HCAs.

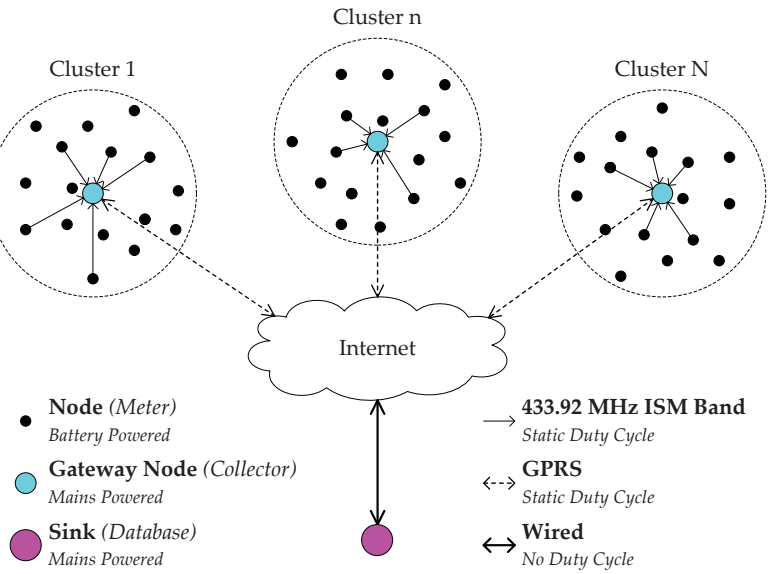


Figure 7.12: Brunata's AMR network topology.

7.3.1 The network of the case study

Brunata's existing system constitutes an hierarchical WSN, as shown in Figure 7.12. Tier-1 is subdivided into clusters of HCAs (i.e.sensor nodes) mounted onto radiators, forming a single-hop network to the collectors. Tier-2 is composed of collectors (i.e. gateway nodes), where a single gateway node is assigned to each cluster. The gateway nodes are wired to each other and connected to the internet over a General Packet Radio Service (GPRS) link. Lastly, tier-3 is a mixture of workstations, servers, and databases (i.e. sinks) that are distributed among a large geographical area.

In addition to HCAs, the gateway nodes are also energy-constrained devices. In an energy harvesting context, gateway nodes are mounted on hot water pipes in buildings and, therefore, have a higher power budget estimated between 100μW to 1mW [121].

The link between the EH-HCA and the gateways is a link where both the senders and the receivers are on duty cycle. To support this link, Brunata uses a simplified adaptation of the ALOHA protocol [2], called IMR+, that only supports a unidirectional, single-hop, single channel AMR network. The sender nodes (i.e. HCAs) only contain a radio transmitter, while the receiving gateway node only contains a radio receiver. Both the senders and receiver use a static duty cycle. Figure 7.13 shows an overview of this approach. In IMR+, random channel access is used to prevent collisions caused

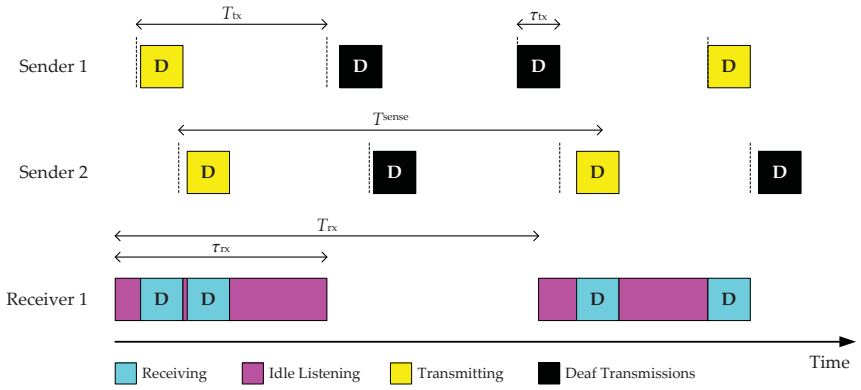


Figure 7.13: IMR+ communication model.

when senders coincidentally synchronize. Due to this simple collision avoidance technique, the small payload sizes used by Brunata, and the ultra-low duty-cycling of the senders, the probability that a collision will happen is quite low. Nevertheless, several significant sources of energy waste exist: *a)* Senders continuously transmit data even though the receiver is in the sleep state. *b)* In sparse networks, the receivers spend most of the active period listening for data, but not receiving any. *c)* Senders that are within range of more than one gateway node will broadcast data to all the receivers.

The metering industry prefers to use MAC schemes that are similar to ALOHA, such as the Wireless M-Bus [15] Mode-C1 or IMR+ in AMIs. The main motivation lies in the perceived belief that the simplistic nature of ALOHA-based MAC schemes outperforms any other MAC scheme. In the next sections, we will analytically compare IMR+, as representative of the "industrial" ALOHA-based MAC schemes, with ODMAC. For this purpose, we model ODMAC and IMR+ with respect to the particular characteristics of the case study.

7.3.2 ODMAC and IMR+ Models

ODMAC and IMR+ are modeled and analytically compared. The properties of the channel are considered to be the same for both schemes and the models do not consider retransmissions due to channel errors. Furthermore, nodes are considered to transmit only a single packet within a duty cycle period, and the packet size is considered to be constant for all transmissions over all nodes. In a single-hop topology, nodes do not relay data to each other. As a result, the model of a MAC scheme can be separated into the sender and receiver in a single cluster. The analysis takes into account the available power for the sender and receiver from the energy harvesting system described in [121].

Lastly, none of the protocols incorporates active collision avoidance mechanisms (such as AB) and both rely solely on random channel access.

7.3.2.1 IMR+ Model

Figure 7.13 shows the model used in the analysis of IMR+. Starting from the sender, the duration of a single transmission τ_{tx} is the time the channel will be used by a single node and can be expressed by (7.7), where L is the packet size in bytes and R is the transmission bit rate in bits per second.

$$\tau_{\text{tx}} = \frac{L \cdot 8}{R} \tag{7.7}$$

Equation (7.8) models the duty cycle period T_{tx} of a single packet transmission in seconds for a given available amount of power $P_{\text{sender}} \in \left[P^{\text{min}}_{\text{sender}}, P^{\text{max}}_{\text{sender}}\right]$ in watts. P_{tx} is the power consumed by the radio during transmission in watts. P^{S} is the power consumed by the radio to enter the active state from the sleep state in watts, and T^{S} is the time it takes in seconds. T^{S}_{tx} is the time it takes for the radio to begin transmission in seconds.

$$P_{\text{sender}} = \frac{P^{\text{S}} \cdot T^{\text{S}} + P_{\text{tx}} \cdot (T^{\text{S}}_{\text{tx}} + \tau_{\text{tx}})}{T_{\text{tx}}} \tag{7.8}$$

Collisions can occur when two or more nodes transmit at the same point in time. The probability of a collision $P(\text{c})$ depends on the number of nodes in the cluster (N).

$$P(\text{c}) = (N-1) \cdot \frac{\tau_{\text{tx}}}{T_{\text{tx}}} \tag{7.9}$$

Given the required probability $P(\text{s})$ of successfully delivering at least one out of n transmissions, the number of transmissions required to guarantee that at least one out of n transmissions is successfully delivered, collision-free, is:

$$P(\text{s}) = 1 - P(\text{c})^n \Rightarrow n = \left\lceil \frac{\log(1 - P(\text{s}))}{\log(P(\text{c}))} \right\rceil \tag{7.10}$$

Since the receiver cannot communicate information back to the senders, it has to be prepared for the worst case scenario time. In order to ensure that at least one valid

packet is received from all the nodes in the cluster, the receiver should listen to the channel for at least time τ_{rx} stated by (7.11), where the worst case duty cycle period of the sender $T_{\text{tx}}^{\text{max}}$ is given by (7.8) for $P_{\text{sender}} \equiv P_{\text{sender}}^{\text{min}}$.

$$\tau_{\text{rx}} = T_{\text{tx}}^{\text{max}} \cdot n \tag{7.11}$$

The receiver duty cycles to ensure operation with the available power, which is denoted as $P_{\text{receiver}} \in \left[P_{\text{receiver}}^{\text{min}}, P_{\text{receiver}}^{\text{max}}\right]$ in watts. The duty cycle period T_{rx} can be calculated using (7.12). P_{rx} is the power consumed by the radio during reception in watts. T_{rx}^{s} is the time it takes for the radio to enter the receive mode in seconds.

$$P_{\text{receiver}} = \frac{P^{\text{s}} \cdot T^{\text{s}} + P_{\text{rx}} \cdot (T_{\text{rx}}^{\text{s}} + \tau_{\text{rx}})}{T_{\text{rx}}} \tag{7.12}$$

The sensing period T^{sense} represents how often a new measurement can be made by a node. Since within a single duty cycle period, the receiver has at least one valid packet from all the nodes in the cluster, the sensing period is the same as the duty cycle period of the receiver, $T^{\text{sense}} = T_{\text{rx}}$. The throughput of the receiver ρ, defined as the amount of new measurements received from all the senders per unit time, can be calculated from (7.13).

$$\rho = \frac{N \cdot L \cdot 8}{T^{\text{sense}}} \tag{7.13}$$

7.3.2.2 ODMAC Cluster Model

ODMAC is modeled as shown in Figure 7.14. While the same notations from Figure 7.13 are used, new notations used in this analysis are described. The receiver performs a CCA to ensure that the channel is idle before transmitting a beacon. The time taken to perform a CCA ($\tau_{\text{rx}}^{\text{cca}}$), is specified for the radio that is used. Once the channel is free, the receiver transmits a single beacon frame. The duration of the beacon $\tau_{\text{tx}}^{\text{b}}$ is described by (7.14), where L_b is the beacon size in bytes. After broadcasting a beacon, the receiver continues to listen for a short period of time to receive a response from a sender.

$$\tau_{\text{tx}}^{\text{b}} = \frac{L_b \cdot 8}{R} \tag{7.14}$$

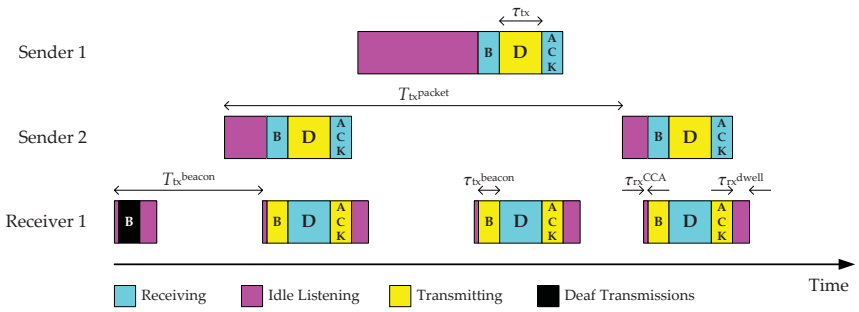


Figure 7.14: ODMAC communication model.

The beaconing duty cycle period T_{tx}^{b}, given by (7.15), depends on the amount of power the receiver is harvesting $P_{\text{receiver}} \in \left[P_{\text{receiver}}^{\text{min}}, P_{\text{receiver}}^{\text{max}}\right]$ in watts. After the sender receives the beacon, it transmits the data immediately, which is then acknowledged by the receiver with another beacon.

$$P_{\text{receiver}} = \frac{P^{\text{s}} \cdot T^{\text{s}} + P_{\text{rx}} \cdot (2T_{\text{rx}}^{\text{s}} + \tau_{\text{rx}}^{\text{cca}} + \tau_{\text{tx}}) + 2P_{\text{tx}} \cdot (T_{\text{tx}}^{\text{s}} + \tau_{\text{tx}}^{\text{b}})}{T_{\text{tx}}^{\text{b}}} \tag{7.15}$$

The transmission duration of a single packet is given by (7.7). When the sender has data to exchange, in the worst case, it has to wait for a full beacon period before receiving a beacon from the receiver. It then immediately transmits the data, and receives an acknowledgment from the receiver. The duty cycle period of the sender T_{tx}^{p} is given by (7.16). The duty cycle period of the sender depends on the power budget $P_{\text{sender}} \in \left[P_{\text{sender}}^{\text{min}}, P_{\text{sender}}^{\text{max}}\right]$ in watts.

$$P_{\text{sender}} = \frac{P^{\text{s}} \cdot T^{\text{s}} + P_{\text{rx}} \cdot (2T_{\text{rx}}^{\text{s}} + T_{\text{tx}}^{\text{b}} + \tau_{\text{tx}}^{\text{b}}) + P_{\text{tx}} \cdot (T_{\text{tx}}^{\text{s}} + \tau_{\text{tx}})}{T_{\text{tx}}^{\text{p}}} \tag{7.16}$$

A collision occurs at the receiver in ODMAC, when two or more nodes that has data to transmit, wake up for the same beacon. Since beacons form time slots for communication, these senders collide when transmitting data after receiving the same beacon. The

Table 7.2: Model parameters.

L_b	18 bytes	R	153600 bps
P^{s}	1.02mW	T^{s}	5.8ms
P_{tx}	132mW	$T^{\mathrm{s}}_{\mathrm{tx}}$	0.5ms
P_{rx}	11.55mW	$P^{\mathrm{max}}_{\mathrm{sender}}$	10μW
$P(\mathrm{s})$	99.99%	$P^{\mathrm{min}}_{\mathrm{sender}}$	1μW
$T^{\mathrm{s}}_{\mathrm{rx}}$	0.5ms	$P^{\mathrm{min}}_{\mathrm{receiver}}$	100μW
$\tau^{\mathrm{cca}}_{\mathrm{rx}}$	100μs	$P^{\mathrm{max}}_{\mathrm{receiver}}$	1000μW

probability $P(\mathrm{c})$ of such an event happening can be expressed by (7.17).

$$P(\mathrm{c}) = (N-1) \cdot \frac{T^{\mathrm{b}}_{\mathrm{tx}}}{T^{\mathrm{p}}_{\mathrm{tx}}} \tag{7.17}$$

As described by (7.10), in the worst case, a sender has to transmit data for n times to ensure that with a probability $P(\mathrm{s})$, at least one transmission is successful. The sensing period of a sender, which is the shortest time duration a sender has to wait before performing a new measurement, is represented by (7.18).

$$T^{\mathrm{sense}} = T^{\mathrm{p}}_{\mathrm{tx}} \cdot n \tag{7.18}$$

The throughput of the receiver can be calculated by (7.13).

7.3.3 Analytical Comparison

The models for IMR+ and ODMAC are used to compare performance of both schemes using MATLAB [79]. Only a relative comparison can be made, since channel errors are not included in the model. However, the models are sufficient to determine the advantages and disadvantages of the two MAC schemes. The harvested and consumed power levels used in the analysis are described in Table 7.2. The range of power levels harvested from the heat of the radiator by the senders are based on the experiment described in [121]. The range of power levels harvested from the heat of hot water pipes in buildings by the receiver are based on the TE-CORE7 [81]. The power consumption levels of the radio are based on the SX1212 transceiver from Semtech [103].

The impact of harvested power on how often a new measurement can be performed is shown in Figure 7.15. Since the receiver cannot communicate any information back

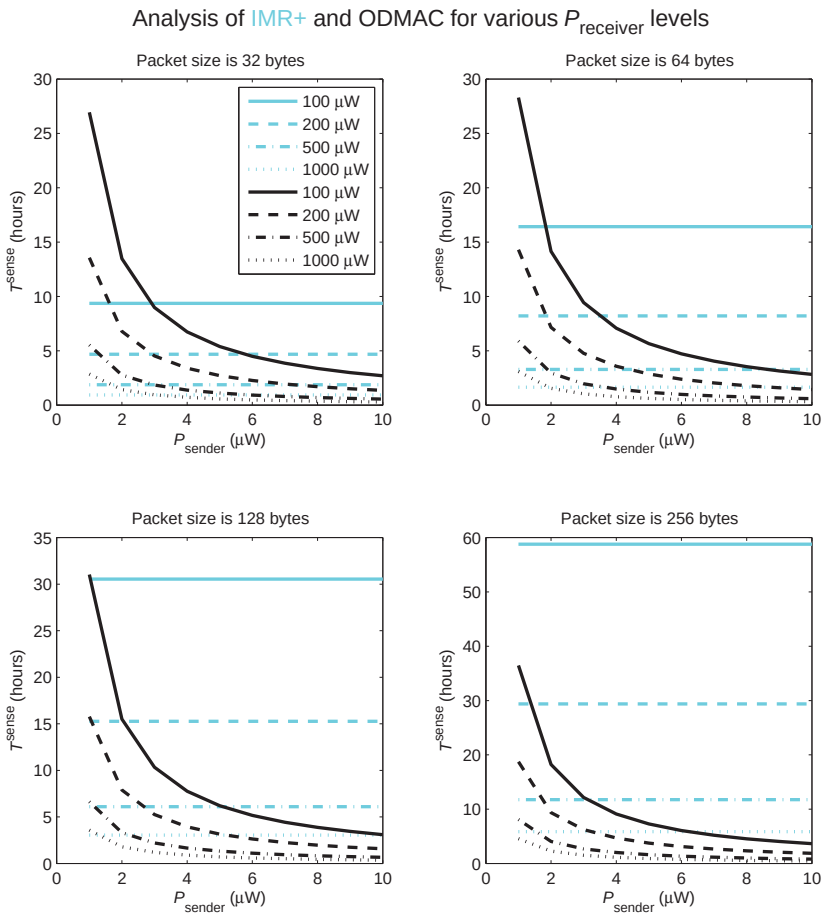


Figure 7.15: Impact of harvested power on the measurement period.

to the sender in IMR+, it has to be designed for the worst case. The receiver cannot efficiently use the power available to increase its performance, since the listening duration should be long enough to ensure that at least one valid packet will arrive successfully from all the senders. Furthermore, IMR+ cannot be used in an EH-WSN application where senders can afford to completely shut down when there is no energy to harvest, because the receiver would have to remain constantly active. In contrast, ODMAC demonstrates its ability to dynamically adjust to the energy harvested from its environment. For very low harvested power, ODMAC sacrifices the frequency of measurements to keep the network stable. As soon as the harvested power increases, the sensing period reduces exponentially, far outperforming IMR+. Another observation of the analysis is the impact of the packet size. In IMR+, the sensing period is

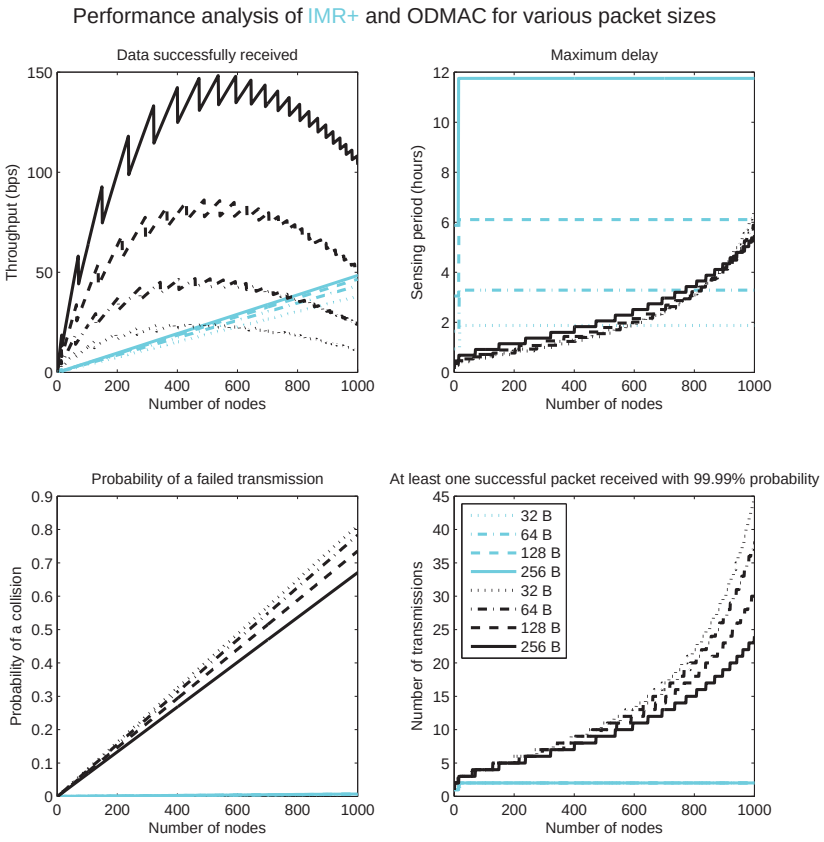


Figure 7.16: Best case: senders and receiver harvest the maximum power.

severely affected by the increase in the packet size, where as ODMAC is more resilient to the increase in packet size while still maintaining the same adaptivity.

The optimal network performance is achieved when both senders and receivers are able to harvest the maximum amount of power from the energy source, as shown in Figure 7.16. The performance of ODMAC drops as the network becomes more dense, when the sender and the receiver are fully active. This is due to the linearly increasing probability of collisions. IMR+ is more resilient and robust in this scenario. However, even with the problem of scalability, ODMAC outperforms IMR+ in a sparse network, while in a dense network it still maintains a faster sensing frequency. While it was previously observed that ODMAC gracefully handles an increase in packet size, it can now be seen that an increase in packet size actually benefits ODMAC significantly, far outperforming IMR+ for large packet sizes. The receiver is flooded when senders are fully active and the receiver is harvesting the lowest amount of power. This scenario

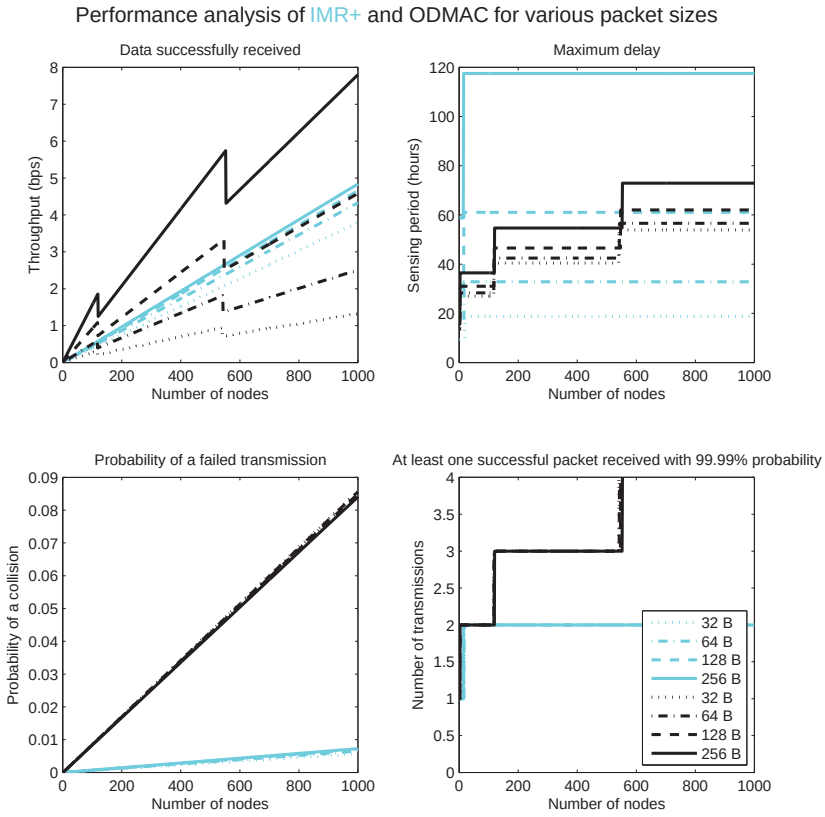


Figure 7.17: Worst case: senders and receiver harvest the minimum power

shows similar characteristics to the best case, but with a very low throughput.

The network has the worst performance when the amount of harvested energy is the lowest for both senders and receivers. Such a scenario is shown in Figure 7.17. For smaller packet sizes ODMAC performs worse than IMR+. However, for large packet sizes, ODMAC outperforms IMR+, and maintains a higher sensing frequency under all circumstances. Furthermore, like IMR+, ODMAC remains scalable and robust.

7.4 Evaluation Summary

In this chapter, we compared ODMAC with two state-of-the-art MAC protocols that are widely used in either the academic or the industrial world.

In the first study, we compared ODMAC with a representative and widely-used MAC protocol from the sender-initiated asynchronous paradigm of communication, namely X-MAC [13]. The analytical results suggest that ODMAC can be tuned to consume less energy than X-MAC. Hence, ODMAC is more suitable in cases of limited environmental energy and the cases where the application requires the system to operate at the duty cycle that provides the minimum energy consumption (e.g. applications that prioritize throughput). On the other hand, X-MAC can provide better performance for delay-sensitive applications in environments where the energy input is sufficiently high. Adjusting several parameters of the system can increase or decrease the performance of the two paradigms; however, the basic trend remains the same.

In the industrial case study, we compared ODMAC with IMR+, the protocol used by Brunata's commercial network. The analytical results show that the simplicity of IMR+ makes it very scalable and robust. However, it is highly unsuitable for energy harvesting applications due to its inability to dynamically manage its resources to improve the performance. Since ODMAC is able to dynamically manage its resources to achieve a maximum performing state for a given amount of energy and it outperforms IMR+. In dense networks, ODMAC is shown to suffer from high number of collisions, making it less scalable and motivating its active collision avoidance extension (see AB in Section 3.4), which is omitted in this study. On the other hand, senders in IMR+ cannot be enriched with an active collision avoidance mechanism due to the lack of a receiver, which constitutes CCA impossible.

An interesting observation from the analysis demonstrates the benefits of buffering. Buffering can further reduce the transmission period. Senders are able to perform sensing tasks and store the packets in a buffer, while the MAC scheme transmits aggregated data packets. As shown from the analysis, IMR+ cannot benefit from buffering, as transmitting more data increases the probability of collisions and the scheme will suffer from poor performance. On the other hand, ODMAC is able to utilize the benefits of buffering efficiently, without a large impact from additional collisions.

CHAPTER 8

Implementation and Testbed Experiments

8.1 Evaluation Overview

In this chapter, we provide an implementation of ODMAC for eZ430-rf2500 wireless sensor nodes by Texas Instruments (Section 8.2) and we conduct experiments (Section 8.3) that demonstrate sustainable operation and evaluate the performance of collision avoidance and traffic differentiation with AB (see Section 3.4). Section 8.4 summarizes the results of the evaluation.

8.2 Firmware Implementation

Our design is based upon the holistic approach, which claims that the whole system should be designed and function as a whole, rather than being organized in layers. This approach sacrifices the versatility of the system toward the efficient use of resources, as all parts of the system, from the hardware to the firmware (i.e. protocols and system services), need to be specifically designed for the desired application. As a result, the implementation of ODMAC constitutes integral part of a complete firmware that

also implements power management, routing and application-related functionalities. The firmware implements a subset of the features of ODMAC that are presented in Chapter 3.

The protocol was implemented on the eZ430-rf2500 wireless development platform [115] by Texas Instruments. The sensor nodes consist of an MSP430 Microcontroller Unit (MCU) and a CC2500 radio, operating in the 2.4 GHz band. In addition to batteries, the nodes can be powered by external energy harvesting boards. In particular, we use Cymbet's CBC-EVAL-10 [18] solar energy harvester board and CBC-EVAL-09 [19] general energy harvester board, that can harvest energy from various sources. The boards store the harvested energy into embedded batteries (100 μAh capacity). The boards can also accommodate external rechargeable batteries for scenarios that require larger energy buffers.

8.2.1 ODMAC as a Finite State Machine

The heart of ODMAC is implemented as an FSM, as shown in Figure 8.4. Its functionality is mainly based upon two routines, namely *Send* and *Receive*. Unless one of these two handlers is invoked, ODMAC is in sleeping state and the radio is turned off.

The *Send* routine generates and formats a packet around the payload (i.e. the result of a sensing operation). When the packet is ready, the radio is switched on into listening mode and the state machine awaits for an interrupt signaling the reception of a beacon. Different packet types might be received while waiting for a suitable beacon. While non-beacon packets are simply discarded, all beacons are evaluated. Upon the signaling of the first suitable beacon, ODMAC continues its execution and the data packet is transmitted. At the end of a packet transmission, the radio is switched back off.

The *Receive* handler is invoked during the forwarding duty cycle. In particular, it generates and broadcasts a beacon packet. At this point, the radio is switched into listening mode and the protocol awaits for a data packet for a defined amount of time. If no incoming data is received during this period, the radio is set back to sleep mode and the routine ends. On the other hand, upon receiving a data packet, the information contained is extracted and the radio set back to sleep mode. In order to forward the newly received packet toward the sink, a new invocation of *Send* is performed.

8.2.2 Implementation of Duty Cycles

Duty cycles are implemented through wake-up interrupts using the low-frequency timer of the MCU. A time quantum is defined. It controls the sleeping time between two sub-

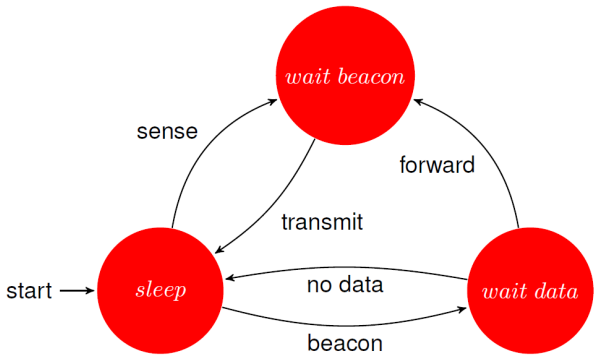


Figure 8.1: ODMAC as a high-level finite state machine.

sequent wake-up events. On top of that, the two independent duty cycles for the sensing and the forwarding tasks are implemented as multiples of the basic time quantum. In each wake-up interrupt the MCU checks if it is time for one of the two tasks, sets up the next wake-up interrupt and goes to the sleeping state. Hence, the duty cycles are controlled by these three configurable parameters.

Additionally, the time quantum is periodically adjusted, by adding a uniformly random number of cycles in $[-2^{r-1}, 2^{r-1}]$ to the defined value. This randomization prevents unfortunate synchronizations and decreases the collisions by enforcing random channel access between different nodes. Even though the period of the time quantum randomization can be individually configured, it is currently set to the period of the sensing tasks. The level of the randomization, r, can be configured in accordance to the desired behavior.

While in the sleep state, the MCU is configured to Low Power Mode 3 (LPM3), in which only the auxiliary low-frequency oscillator ($12KHz$), used to schedule the interrupts, is active. In LPM3, MSP430 consumes less than 1 μA at 1 MHz [117].

8.2.3 Integration of Layer-based Anycast Routing (LAR)

Additionally, we implemented and incorporate inside the routines of ODMAC the LAR algorithm (see Section 3.6.4). Specifically, the sink node initializes its layer to 0, while all the sensor nodes initialize their layer to 99 which represents that the nodes are disconnected from the network. Unless they are disconnected from the network, nodes advertise their layer through their beacons.

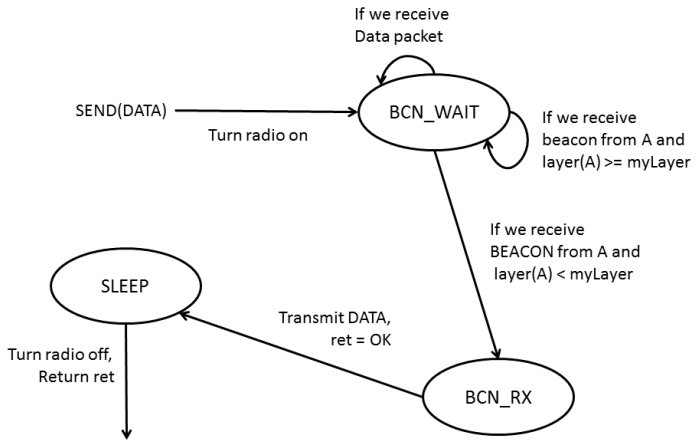


Figure 8.2: The finite state machine describes the operation of ODMAC with no collision avoidance.

All nodes update their layer during the beacon evaluation of the *Send* routine. In particular, there are four distinct cases. (i) A sender may receive a beacon that advertises a layer that is greater or equal than its own layer. In this case, the beacon is discarded and the node continues listening the channel. (ii) A sender may receive a beacon that advertises a layer that is lower than its own layer by exactly 1. In this case, the beacon is marked as suitable and an interrupt is generated that signals the data packet transmission. (iii) A sender may receive a beacon that advertises a layer that is lower than its own layer by more than 1. In this case, the sender updates its layer to 1 more than the layer advertised of the beacon. Then, the beacon is marked as suitable and an interrupt is generated that signals the data packet transmission. (iv) A sender may not receive any beacon within a predefined time interval. In this case, it updates its layer to 99 and considers itself disconnected from the network.

8.2.4 Implementation of Collision Avoidance

The protocol supports three modes for collision avoidance, namely *No Collision Avoidance (NOCA)*, *CB* and *AB*. NOCA does not implement any additional functionality and, therefore, relies only on the duty-cycle randomization for collision avoidance (Figure 8.2). CB is implemented by adding a random delay between the reception of a suitable beacon and the transmission of the data (Figure 8.3).

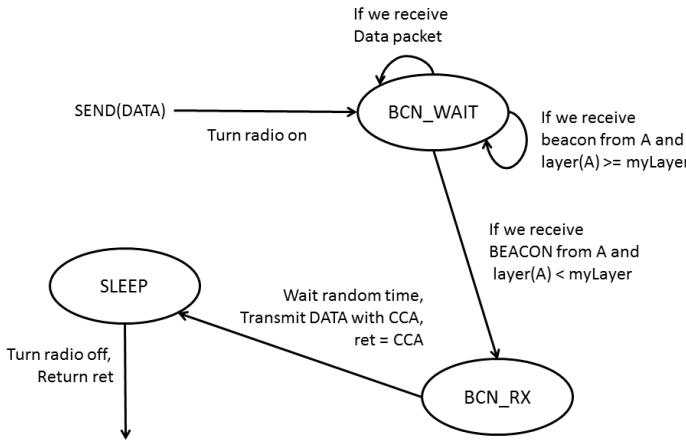


Figure 8.3: The finite state machine describes the operation of ODMAC with Constant Backoff (CB).

AB extends the *Send* routine as follows. The ABR control packet is implemented similarly to a beacon. Specifically, in the ABR, we include the layer that indicates the group of beacons that the sender is waiting for, to any potential contenders that happen to be awake. After a successful CCA the transmission of the ABR follows. Then, the radio is switched to listening mode and the sender begins to listen for a beacon. Listening is interrupted either by the reception of a suitable beacon or by the reception of an ABR that advertises the same layer as the layer of the sender. In the former case, data transmission follows normally. In the latter case, the routine returns and indicates a backoff. The state machine in Figure 8.4 summarizes the operation of AB as part of the ODMAC protocol. Figure 8.5 summarizes the behavior of a sender while waiting for a beacon.

It should be noted that the *Send* routine performs one attempt to transmit the packet. In case of backoff, the higher layer is free to decide at which point in the future will attempt again to transmit the same packet.

For the traffic differentiation services of AB, we extend the implementation by adding a priority bit in the header of ABR control packets. The priority bit indicates if the data packet is classified as *High Priority* or *Best Effort*. When a sender that waits for a beacon, receives another ABR packet, it compares its local priority bit with the received priority bit. If and only if the local data packet is classified as *High Priority* and the received ABR indicates a *Best Effort* data packet, the sender retakes the channel by

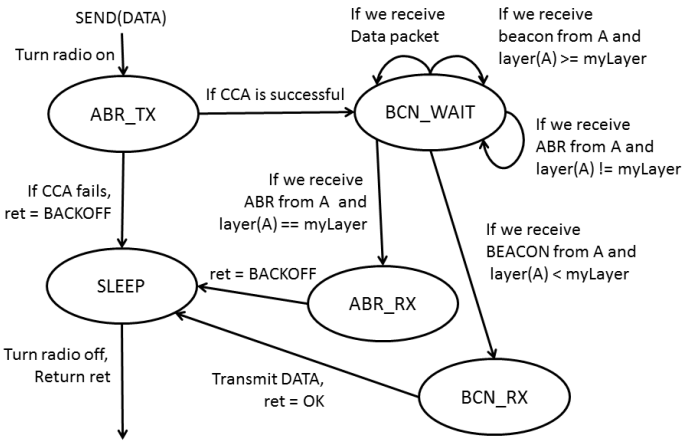


Figure 8.4: The finite state machine describes the operation of ODMAC with Altruistic Backoff (AB).

invoking the *Send* routine again.

8.2.5 Packet Errors

The implementation does not include any mechanisms that react to lost packets due to channel errors, such as acknowledgments and retransmissions. Instead, it numbers the packets with an 8-bit sequence number that is included in the payload. The sequence number is used by the sink node to detect if and when a packet has been lost.

8.2.6 Security Extensions

Link-layer authentication and encryption services, inspired by TinySec [60], have also been designed and implemented. The security suite provides four modes: *No security, Authentication, Encryption, Both* allowing to choose among them on a per-message basis. Both confidentiality and integrity are provided through the same encryption primitive, namely *Skipjack* [1]. The implementation of Skipjack is based on the open source implementation for OpenBSD, and it is changed accordingly to meet the memory constraints of MSP430. Encryption is always performed in CBC mode [105]. Authenti-

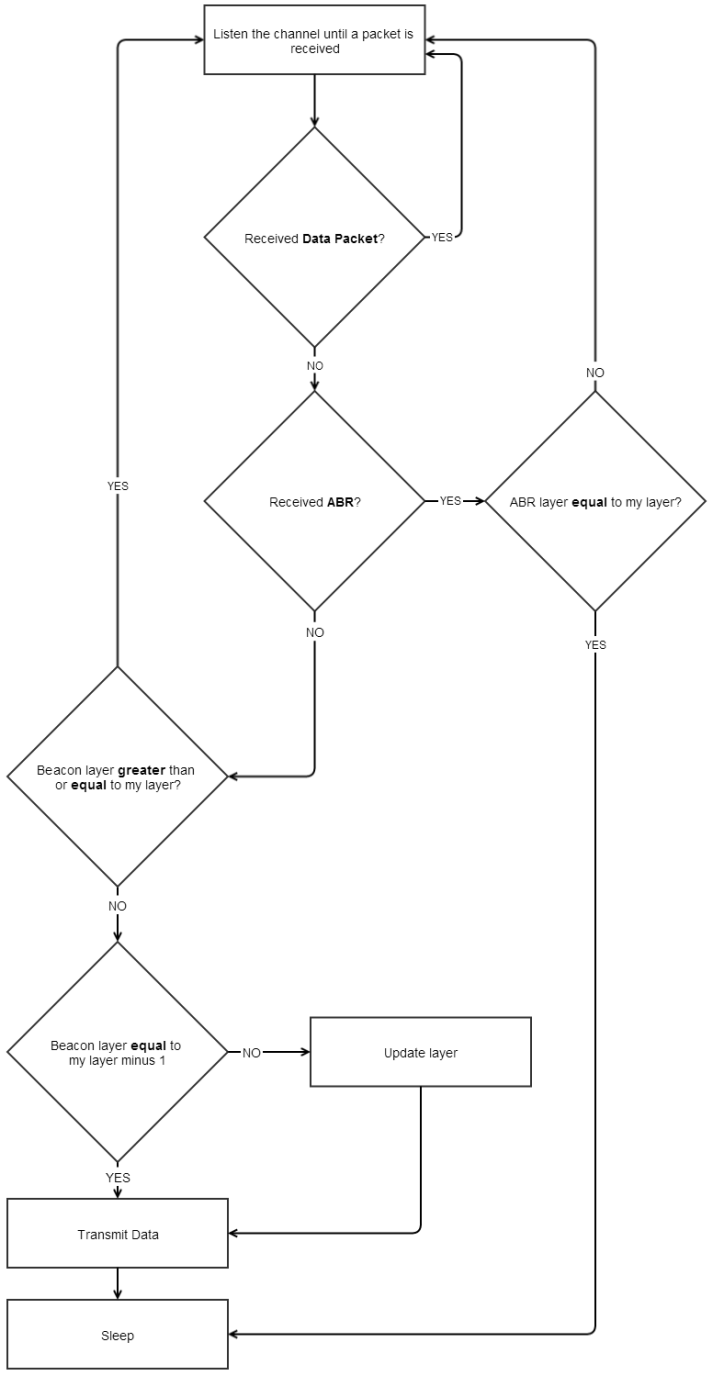


Figure 8.5: The behavior of a sender.

Table 8.1: Packet types (TYPE) in options.

Options bits	Packet Type
00	Beacon
01	Data Packet
10	ABR
11	*reserved*

Table 8.2: Security modes (SEC) in options.

Options Bits	Security Mode
00	No Security
01	Encryption
10	Authentication
11	Encryption + Authentication

cation appends to the packet a 4-byte footer that contains the message authentication code. Any authenticated packet whose code is not verified correctly is dropped. In case both encryption and authentication are enabled, encryption is performed first and the message authentication code is computed on the cipher-text.

8.2.7 Packet Formats

Both beacons and data packets have a 8-bit options field in their header (OPT). The option field is a bitmap that specifies how each packet should be handled by its receiver. The two least significant bits specify the type of the packet, as shown in Table 8.1. The next two least significant bits in the options specify the security mode for the specific packet, as shown in Table 8.2. The forth least significant bit in the options is reserved for the yet unimplemented feature of acknowledgments as shown in Table 8.3. The fifth least significant bit is indicating the priority class of the packet (see Section 3.4), as shown in Table 8.4. The two most significant bits in the options are reserved for future extensions. The options byte is summarized in Figure 8.6.

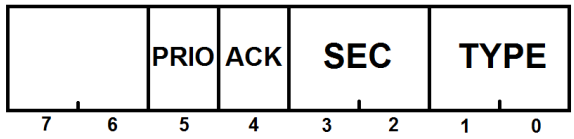


Figure 8.6: The options byte (OPT) format.

Table 8.3: Acknowledgments (ACK) in options.

Options Bit	Acknowledgment
0	No data packet successfully received
1	Data packet successfully received

Table 8.4: Priorities for traffic differentiation (PRIO) in options.

Options Bit	Priority Class
0	Best Effort
1	High Priority

The payload of a beacon or an ABR frame consist of an 1-byte field that specifies the layer, which is used to assess the suitability of the beacon or the need for backoff in case of ABR. The payload of a data packet is 20 bytes in total and contains information such as the identification number of the node, the sequence number of the data packet, the measured temperature of the internal sensor of the MCU and other statistical information. In case of authentication is enabled, packets also have a 4-byte footer that includes the message authentication code. Figure 8.7 summarizes the packet format.

8.2.8 Energy Awareness

To incorporate energy awareness in the duty cycles, the Analog-to-Digital Converter (ADC) of the MCU is switched to channel 0 (input pin $A0$) and reads the voltage of the energy buffer. Before turning the radio on, in both communication routines (*Send* and *Receive*), the firmware measures the voltage from pin $A0$ and proceeds only if its value is above a configurable threshold. This mechanism dynamically alters the duty cycles in such a way that the radio is never switched on unless there is available energy to support it, as introduced in Figure 1.6.

To use this feature, a hardware modification is required. The positive side of the energy buffer needs to be wired to pin $A0$ and the negative side of the energy buffer needs to be wired to the ground.

Figure 8.7: The packet format. MAC refers to the message authentication code.

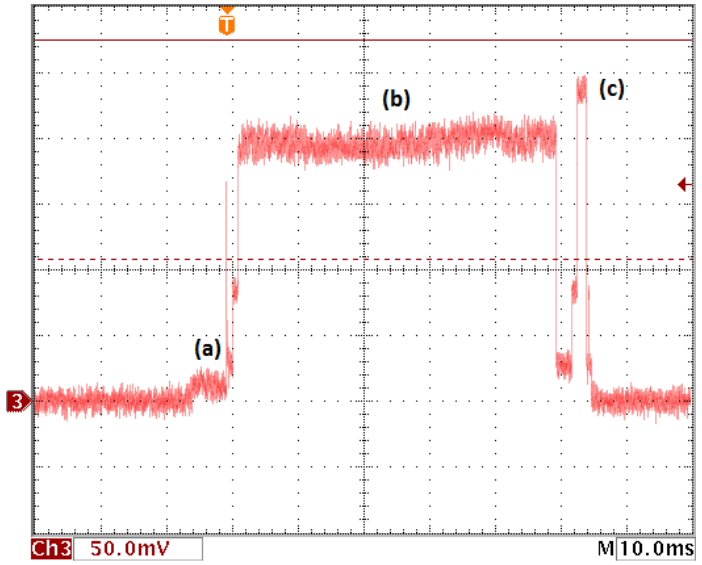


Figure 8.8: Consumption of a typical duty cycle. The current drain is obtained by dividing the shown voltage by the shunt resistor's value (10 Ω). The activity cycle consists of the following actions: a) sensing and packet generation, encryption and authentication, b) waiting for a beacon from the receiver, c) transmiting the packet. Power consumption is dominated by the time the radio spends waiting for a beacon, i.e. idle listening.

8.3 Experimental Evaluation

In this section, we experimentally evaluate ODMAC in a testbed composed of eZ430-rf2500 sensor nodes. The experiments focus on a single link.

8.3.1 Current Profile

Each sensor node is programmed to periodically interrupt its sleeping to execute an active period, which consists of the following actions: (i) sense the MCU temperature using the internal temperature sensor, (ii) generate a packet, (iii) encrypt and authenticate the packet, (iv) wait for a beacon from the receiver, (v) transmit the packet. The consumption of a typical activity period is shown in Figure 8.8. Specifically, the figure shows the voltage of a 10 Ω shunt resistor, connected between the load and the power source. In the figure, one can clearly notice the time the node is listening for a beacon, which follows some initial MCU activity that includes using the node's temperature

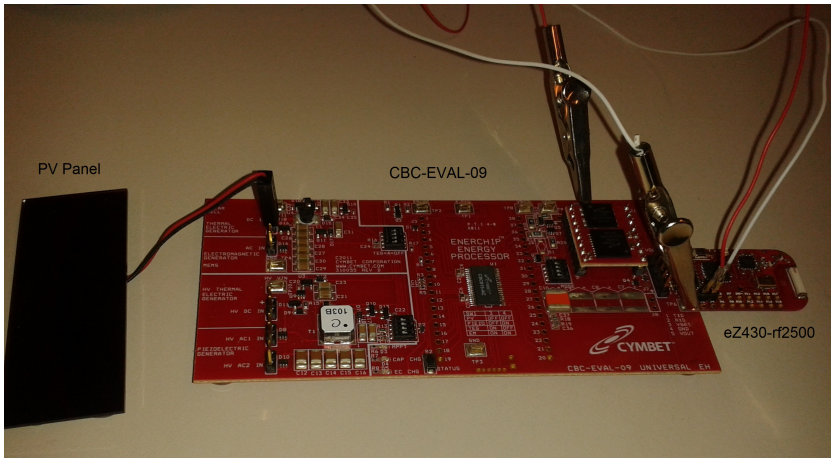


Figure 8.9: The energy harvesting sensor node (eZ430-rf2500) is powered by a photovoltaic panel connected to a CBC-EVAL-09 energy harvester board.

sensor, the ADC and an Light Emitting Diode (LED). After the beacon reception, it is possible to see the consumption spike related to the packet transmission. The example indicates that the main source of power consumption comes from the time the radio spends in listening mode, waiting for a beacon. Hence, it highlights the significance of idle listening mitigation mechanisms, such as Opportunistic Forwarding and AB.

8.3.2 Integration with the Energy Harvester

The energy harvesting sensor nodes are powered by a Photo-Voltaic (PV) panel connected to a CBC-EVAL-09 energy harvester board, as shown in Figure 8.9. The harvester in use is designed around factory specifications that support relatively short high-consumption activity periods (e.g. whenever the radio is on). The energy accumulated in the solid state batteries of the board is used to charge the following stage, composed of a 1000 μF capacitor that is then used as the final energy output. Such component is designed to handle long drains of low current, but short pulse current drain would fully deplete its charge, without giving time to the batteries to recharge it. According to [20], the embedded solid state batteries cannot charge the capacitor in less than approximately 10 seconds. Depleting the capacitor resets the node and triggers a protection mechanism, that disconnects the load until the capacitor is fully charged. Empirically, we found that the capacitor can support activity periods with duration in the order of tens of ms (up to ≈ 150 ms).

When using the specific energy harvester, we need to wire the positive side of the output

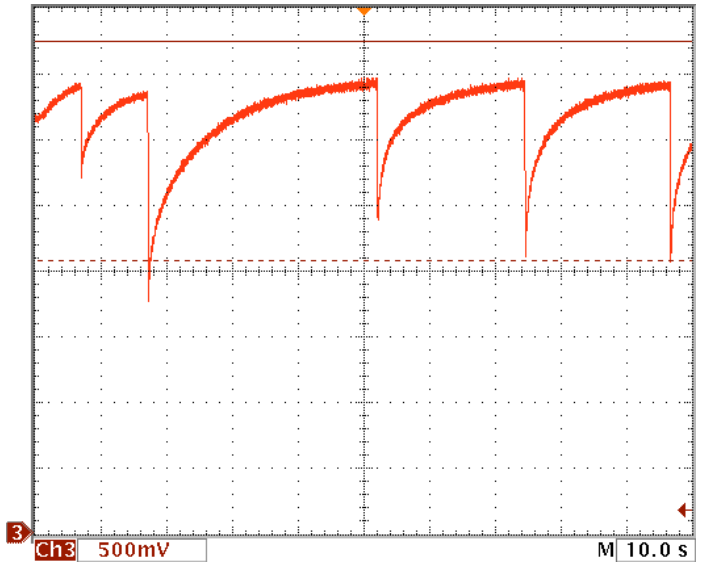


Figure 8.10: The output capacitor voltage demonstrates a typical series of activity periods (duty cycles).

capacitor of CBC-EVAL-09 to the input pin $A0$ of the MCU. However, the voltage of the capacitor is not in the range that the ADC is able to read. To bypass this issue, we use a custom circuit that transforms the signal before feeding it to the ADC.

The duty cycle of the energy harvesting sender is configured as a multiple of the sensing duty cycle, based on the state of the capacitor. Specifically, we set the wake-up interrupts every 12048 cycles of the low frequency oscillator ($\approx 1\ s$). Given the minimum time required for the capacitor to be charged [20], we check the state of the capacitor every 10 wake-up events ($\approx 10\ s$) and transmit when the voltage across the capacitor is above the empirically found threshold of $3.3\ V$. This solution allows us to dynamically adapt the duty cycle (and therefore the amount of packets sent) according to the amount of energy harvested, making the application energy aware. Figure 8.10 depicts the voltage of the capacitor in a succession of packet transmissions. Observe how the energy required for different transmissions varies with respect to the duration of the listening period, while the time for the capacitor to recover changes accordingly.

8.3.3 Sustainability and Throughput

First, we focus on the case study of applications that prioritize the throughput. Specifically, we focus on a single transmitting node, u, which is part of a single link to a

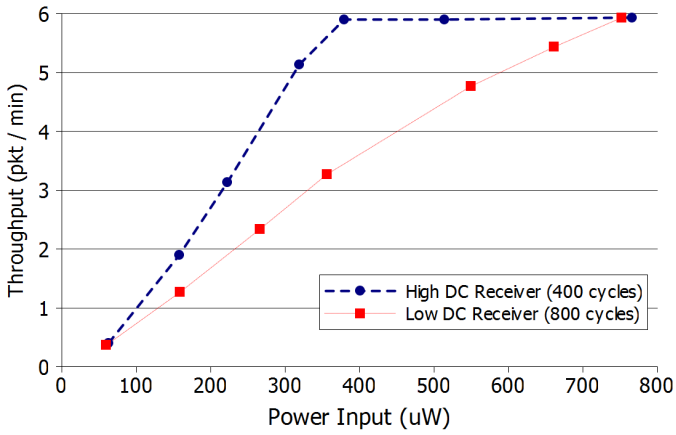


Figure 8.11: Sustainable operation prioritizing throughput for different levels of input power.

receiver node. From the perspective of u, the activity of the receiver is unknown. We consider two identical receivers, one with high and one with low DC. We programmed the sleep time between the transmission of two beacons at 400 cycles for the *High DC Receiver* and 800 cycles for the *Low DC Receiver*, which corresponds to approximately 33 ms and 66 ms respectively. Additionally, we turned off the forwarding duty cycles of u, focusing entirely on the sensing duty cycles. Given this specific configuration and network topology, the average duration of an active period was found to be 43 ms with a standard deviation of 11 ms in the case of the *High DC Receiver* and 61 ms with a standard deviation of 23 ms in the case of the *Low DC Receiver*.

In this setting, we conducted the following experiment. We exposed the energy harvester to different levels of constant input power, by adjusting the distance between the light source and the PV panels, and we measured the amount of packets that the node managed to successfully transmit in 30 minutes. The input power is estimated using the voltage measured across the PV panel and the current measured through a 10 Ω shunt resistor.

Figure 8.11 shows the results of several experiments. All experiments were initiated after the depletion of all the stored energy. Given the fact that the capacitor can not store enough energy for more than very few transmissions, the 30-minute continuous operation demonstrates the sustainability of the node. Furthermore, the excess of harvested energy is used to improve the throughput of the application. As expected, the throughput increases linearly with the amount of available energy, while it is capped by the maximum throughput supported by the energy harvesting board, i.e. 1 transmission every 10 seconds. The difference in throughput, in the cases of the high and low duty

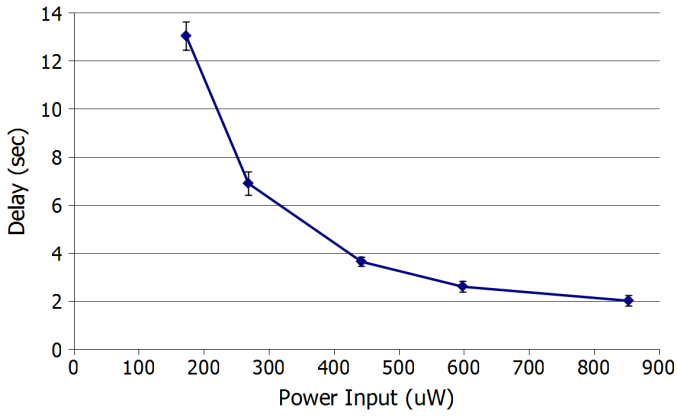


Figure 8.12: Sustainable operation prioritizing link delay for different levels of input power.

cycle receivers, shows how u was able to adapt to different environmental conditions in terms of energy consumption.

8.3.4 Sustainability and Delay

Next, we focus on applications that prioritize short delays. Focusing again on a single link, node u is now the receiver that forwards traffic from a sender. The sender is programmed to transmit data traffic at random times (1 packet per minute on average). The receiver, u, attempts to transmit a beacon every wake-up event ($\approx 1\ s$). Similarly to the previous experiment, the transmission occurs only if the voltage across the capacitor is above the threshold of $3.3\ V$. In this setting, we measure the link delay as the duration of an activity period at a sender node. This approach disregards the propagation delay, which is negligible compared to the other delay sources. Figure 8.12 shows the average link delay of several hundreds of transmissions at several constant power input levels. The error bars indicate the 90% confidence intervals. The 4 to 10-hour continuous operation at each power input demonstrates the sustainability of the node. Additionally, the link delay decreases exponentially with the amount of available energy, while it converges to the delay the corresponds the highest beaconing frequency.

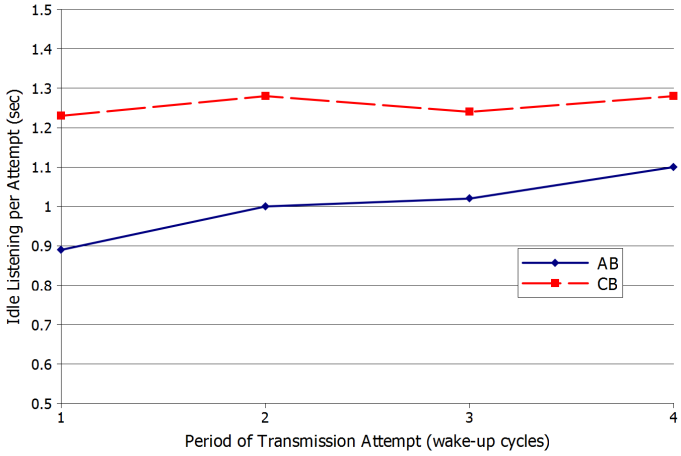


Figure 8.13: Average idle listening per transmission attempt for Altruistic (AB) and Random Backoff with constant (CB) CW. Wake-up interrupts are uniformly randomized after each transmission to enforce random channel access.

8.3.5 Evaluation of Altruistic Backoff (AB)

Lastly, we experimentally evaluate AB by comparing it with CB. We chose to compare AB with the CB variation of RB because the simulations (see Section 5.3) indicate that the variation of the protocol does not affect the idle listening overhead significantly. For CB, we use a constant contention window ($cw = 4$) and a timeslot of 100 MCU cycles ($\approx 100\mu s$).

To measure the idle listening time interval, we use the internal timer unit, which is set to use the low frequency oscillator ($12KHz$) that remains active when the MCU goes into low power (i.e. sleeping) modes. Because of the size of its counter register (16 bits), the timer is able to measure time intervals up to approximately 5.5 seconds. Each node is set to keep the sum of all the time it spent in idle listening since reset and reports the value in every data packet. In addition to that, a sequence number of all the data transmission attempts is also reported in the payload of the data packet. Using the two aforementioned values, we can estimate the average time a node spent in idle listening per data transmission attempt.

For the experiments presented in this section, we use the following testbed. We use a single-hop star topology with a set of battery-powered senders contending to transmit to a single receiver. The contending senders are placed physically close to each other and to the receiver, in order to mitigate any packet losses due to channel errors. The

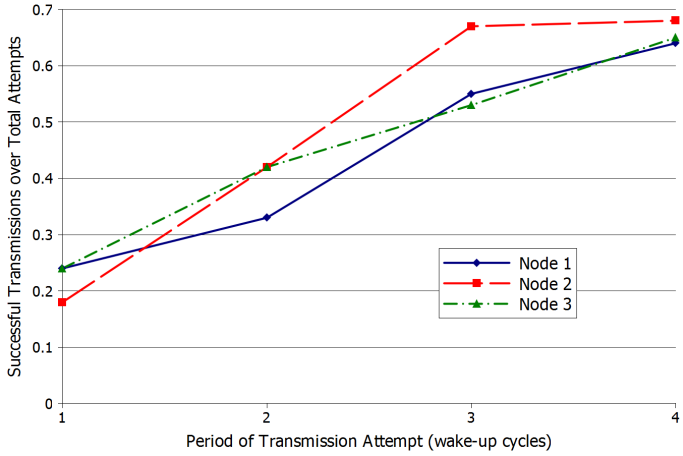


Figure 8.14: The ratio of successful transmissions over the total number of transmission attempts indicates that AB is long-term fair.

receiver node is set to periodically transmit beacons but never generate data of its own. A set of sender nodes are configured to periodically transmit data to the receiver. We use ODMAC's randomization feature to enforce random channel access. In particular, after each data transmission, the wake-up interrupts are randomized over the whole space of the register ($r = 16$). The node then calls the *Send* routine once every sm wake-up interrupts.

In the experiment shown in Figure 8.13, we set the beaconing period of the receiver to 4 seconds and we used 3 contending senders. In the x-axis, we variate the period of a transmission attempt for all the senders in wake-up interrupts, i.e. the sm parameter. The duration of each experiment was 1 hour. The results indicate a similar trend to the respective simulation experiment, shown in Figure 5.3, which verifies the energy consumption improvements of AB. CB follows a similar constant behavior. AB, on the other hand, is spending less time in idle listening and improves as the traffic increases.

Figure 8.14 shows the ratio of successful transmissions over the total number of transmission attempts for the same experiments for AB. The results demonstrate the long-term fairness of the protocol, as the nodes appear to have equal opportunities to take the channel. We can notice that none of the senders is led to starvation and the number of times they took the channel is at the same order of magnitude between the three nodes. The relative difference between the senders is attributed to the duration of the experiment (1 hour). We expect longer experiments to smooth such differences out.

In the next experiment, we fix the period of transmission attempts to 2 wake-up cycles and we variate the number of contending nodes from 1, i.e. no contention, to 4. Fig-

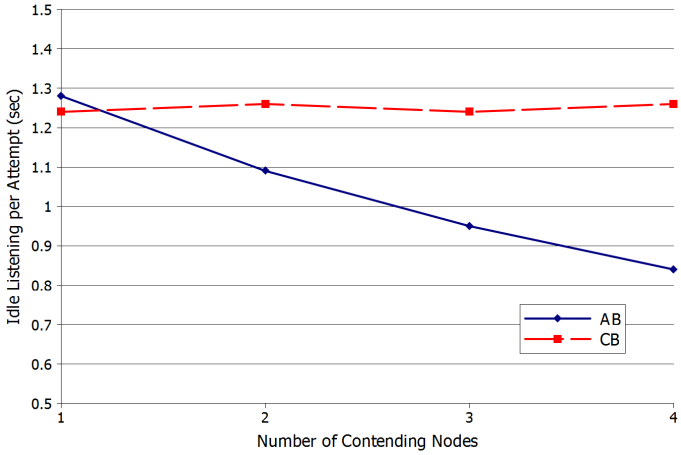


Figure 8.15: Average idle listening per transmission attempt for Altruistic (AB) and Random Backoff with constant (CB) CW for different numbers of contending nodes.

ure 8.15 shows the average time each node spends on idle listening per transmission attempt for the two protocols. The duration of each experiment was 1 hour. The results follow a similar trend to the respective simulation experiment, shown in Figure 5.2. In particular, when no contention the two protocols have similar performance. For the case of CB, the average time spent in idle listening remains constant, being dominated by the time the node waits for a beacon. In the case of AB, on the other hand, idle listening decreases as the contention increases.

Next, we evaluate the long-term fairness of AB in the scenario of contending senders with different traffic generation frequencies. Such scenario has interest in cases of nodes with different forwarding duties or different power resources (e.g. energy harvesting sensor nodes have access to different levels of surrounding energy). The experiment is designed as follows. We use 2 nodes and fix the period of transmission attempts of the first node to 4 wake-up interrupts, while varying the period of the second node from 2 to 4. The duration of each experiment is 2 hours. Figure 8.16 shows the results of the experiment. The triangle-line shows the ratio of the packets generated by Node 1 over Node 2, which increases as the period of transmission attempt of Node 2 increases. Note that, when the nodes have equal periods, the ratio is close to 1. We observe that, despite the fact that the two nodes attempt to use the channel at different frequencies, they maintain equal opportunities to obtain the beacon. The ratio of success full packet transmissions over the total amount of transmission attempts shows a constant behavior.

In the next experiment, we experimentally evaluate traffic differentiation by replicating

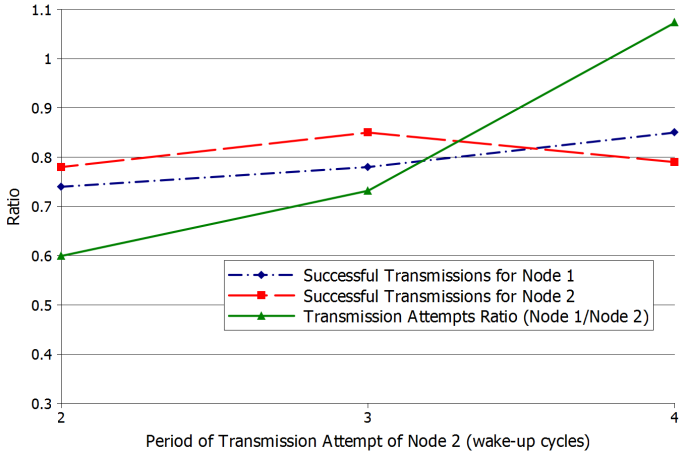


Figure 8.16: The ratio of successful transmissions over the total number of transmission attempts for Node 1 and Node 2 indicates long-term fairness. The period of transmission attempts for Node 1 is fixed to 4 wake-up cycles. The triangle-line shows the ratio of the packets generated by Node 1 over Node 2.

the simulation shown in Figure 5.5. The beaconing period of the receiver is set to 1 second and the period of transmission attempts of the senders is randomized with an average of approximately 3 seconds. Moreover, nodes generate *High Priority* data packets with a probability of $p = 0.05$. Figure 8.17 shows the average ratio of the amount of data packets that take a beacon over the total amount of generated packets, for each priority class. The duration of each experiment is 1 hour. Due to hardware constraints, the experiment was conducted with up to 6 contending nodes. The results validate the simulations and show that as the contention increases, a larger amount of *Best Effort* traffic backs off, giving priority to the *High Priority* traffic.

In the last figure, we validate the simulations by comparing their estimations to the results obtained through the experimental evaluation. In particular, we configure the simulator to the exact same configuration that is used in the testbed experiment presented in Figure 8.15. In the experiment the period of transmission attempts of the senders is set to 2 wake-up cycles that are uniformly randomized over the whole space of the register, leading to an average period of approximately 5.5 seconds. Thus, in the simulator we set period of transmission attempts to 5.5 seconds. The beaconing period of the receiver is set to 3 seconds. Figure 8.18 plots the ratio of the average idle listening per transmission attempt of AB over CB as obtained from the simulation and the testbed experiment. Observe that both simulations and experiment give close results, while the behavior of the protocols follows the same trend. The relative difference indicates that, in the experiments, random access is not as uniformly distributed

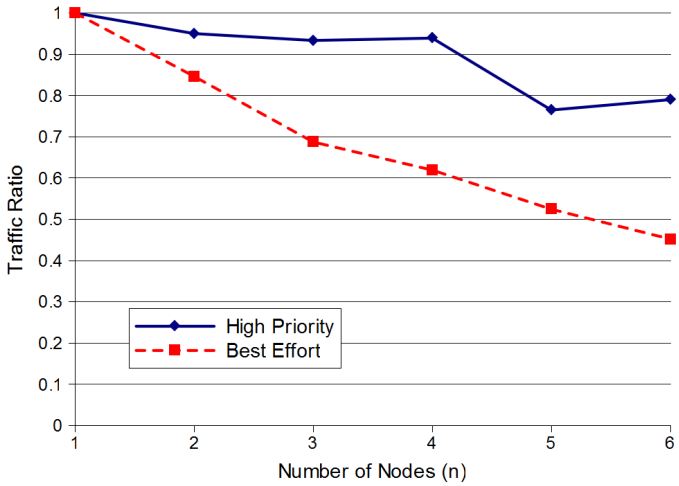


Figure 8.17: The average ratio of the amount of data packets that take a beacon over the total amount of generated packets for each priority class. As the contention increases, the protocol sacrifices *Best Effort* traffic for *High Priority* traffic.

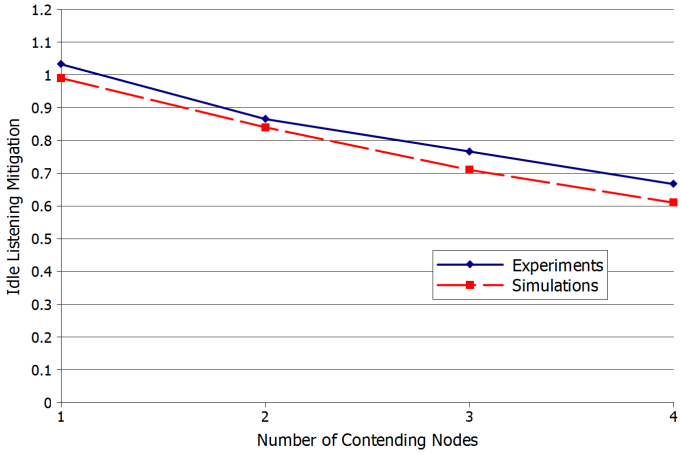


Figure 8.18: Comparison of simulations and experiments. The ratio of the average idle listening per transmission attempt of AB over CB as obtained from the simulations and the testbed experiments.

throughout the interval between two beacons, as assumed in the simulations.

8.4 Evaluation Summary

The presented testbed experiments verify the analysis, presented in Section 4.4, and demonstrate that sensor nodes are able to configure their duty cycle to find a sustainable state of operation and use the available energy to promote the selected performance metric, which can be either throughput or delay.

With regards to AB, the experiments verify the trends that are suggested by the simulations, presented in Section 5.3 and show that AB scales well with both high contention and high traffic and provides equal opportunities for the contending nodes to access the channel (i.e. long-term fairness). Detecting the inevitable collisions before the beacon transmission allows the nodes to resolve the collision before significant amount of energy is wasted in idle listening while waiting for the beacon. Furthermore, AB provides QoS by prioritizing traffic of different urgency. AB is compared to the commonly used collision avoidance mechanism, namely RB, and the results demonstrate the energy savings that can be achieved with AB.

CHAPTER 9

Links with Always-On Receivers

9.1 The case of Links with Always-On Receivers

In this chapter, we will move our attention to links where only the sender duty-cycles while the receiver is always in an active state, as briefly introduced in Section 1.4.1. The chapter, first, presents the development of a prototype energy-harvesting CO2 sensor node that operates with IEEE 802.11 [55], commonly known as Wi-Fi (Section 9.2). Then, we discuss the ambitious idea of using of timing channels in the context of energy-efficient WSN, to encode the measurement in the duration of the sleeping period (Section 9.3). Section 9.4 summarizes the chapter.

9.2 IEEE 802.11 (Wi-Fi) in Wireless Sensor Networks

This section presents a case study investigated in collaboration with WindowMaster A/S. WindowMaster is a company that specializes in the development of building automation applications. The specific project was about the development of a CO2 sensor node that is powered by artificial indoors light. The CO2 measurements indicate how

crowded the room is and are used to automatically control the windows and the ventilation, via a platform that is named NV Comfort [129].

The hardware is based on a prototype circuit developed by WindowMaster. It is composed of an RTX4100 [100] Wi-Fi module and a COZIR Ambient CO2 sensor [17]. RTX4100 consist of an Energy Micro EFM32G [33] MCU and a Atheros AR4100 [97] 802.11n Wi-Fi radio. The circuit is powered by a rechargeable lithium battery that is charged by embedded solar panels through a BQ25504 converter.

9.2.1 Ultra Low-Power Wi-Fi

IEEE 802.11 [55], commonly known as Wi-Fi, defines DCF, a MAC protocol that is based on the CSMA/CA scheme. The typical topological structure is a star where multiple wireless stations are associated with an Access Point (AP) that connects them to the network infrastructure (e.g. Internet).

The MAC protocol defined in IEEE 802.11 does not focus on energy-efficiency, as both the wireless nodes and the AP are active continuously. RTX4100 provides low-power version of Wi-Fi, marketed as Ultra Low-Power Wi-Fi, that incorporates duty cycling in the operation of the wireless stations. Since the AP is continuously active, the establishment of the link does not constitute a particular challenge. The nodes simply follow a sleep-connect-disconnect-sleep cycle. For instance, the provided operating system supports cycles where the wireless node wakes up, connects to the network after associating with the AP, communicates with a server and goes back to sleep.

The key advantage of developing sensing applications with this approach, is the compatibility with existing networks and infrastructures. The use of the Transmission Control Protocol / Internet Protocol (TCP/IP) stack allows the implementation of cloud applications, as the sensor nodes can directly communicate with any computer in the network. Furthermore, the users that have already deployed a WLAN in their building, do not need any additional hardware to support the sensing application. Moreover, the development of plug-and-play sensing applications is possible. On the negative side, IEEE 802.11 and the TCP/IP stack are not optimized for energy-efficiency.

9.2.2 Firmware Overview

The firmware is developed with respect to the particular requirements of the application. The system is required to react quickly to a significant change in the CO2 concentration and to operate in a sustainable manner with the available harvested energy. Secondarily, for statistical purposes, the more measurement are collected, the merrier.

The firmware of the sensor node operates on a basic duty cycle. In the beginning of the cycle, the firmware assesses the energy availability. If this assessment is successful, the firmware continues its operation. In the next step, the firmware activates the CO2 sensor and polls it for a measurement. An assessment of the measurement follows and if it is decided that the specific measurement should be transmitted to the server, the communication procedure begins. At the end, the firmware puts the hardware into sleep mode until the next cycle.

The energy availability assessment is based on the comparison of the voltage of the lithium battery to a configurable threshold. The MCU reads the voltage of the lithium battery through its ADC that is wired to the battery. If the voltage of the lithium battery is below the threshold, the hardware goes to sleep until the next cycle.

The CO2 sensor implements a digital filter to smooth the noise in the CO2 concentration measurements out. In a nutshell, the digital filter calculates a rolling average on the last measurements. It is empirically found that a rolling average of 24 measurements is required to limit the variation of the measurement to less than 5%, in a constant environment. Similarly, a rolling average of 12 measurements is required to keep the noise less than 10%. The CO2 sensor performs one measurement every 0.5 seconds in active mode. Its energy consumption is directly related to amount of time it is in active mode.

To promote the energy-efficiency and meet the requirement for a quick reaction to significant changes, the firmware transmits the measurement only if it is significantly different than the previously reported measurement. This is implemented as a two-level filtering system that works with two threshold levels, as shown in Figure 9.1. In particular, the system keeps the level of the previously reported measurement and it compares it with the current one, calculated as the rolling average of 12 actual measurements from the digital filter of the CO2 sensor. If this difference is bigger than the upper threshold (e.g. $> 10\%$), the measurement is reported. If the difference is smaller than the lower threshold (e.g. $< 5\%$), the measurement is dropped. If the measurement is between the upper and lower threshold, the measurement is kept and reassessed in the next cycle. In the next cycle a new measurement is taken. The new measurement is averaged with the held measurement from the previous cycle and it is transmitted if and only if it is above the lower threshold. This way, the system reports fast any big changes to the CO2 concentration, controlled by the upper threshold. Smaller changes, controlled by the lower threshold, are also reported but with a two-cycle delay.

The communication procedure follows the TCP/IP stack. The firmware turns the radio on and associates with the AP that it was previously associated with. It then executes the Dynamic Host Configuration Protocol (DHCP) protocol to dynamically obtain an Internet Protocol (IP) address. Then, it executes the Address Resolution Protocol (ARP) protocol to find the MAC address of the (local) server. It then establishes a connection to the server. The server device supports two server applications, a web server and a User Datagram Protocol (UDP) server. In case the web server is selected, a

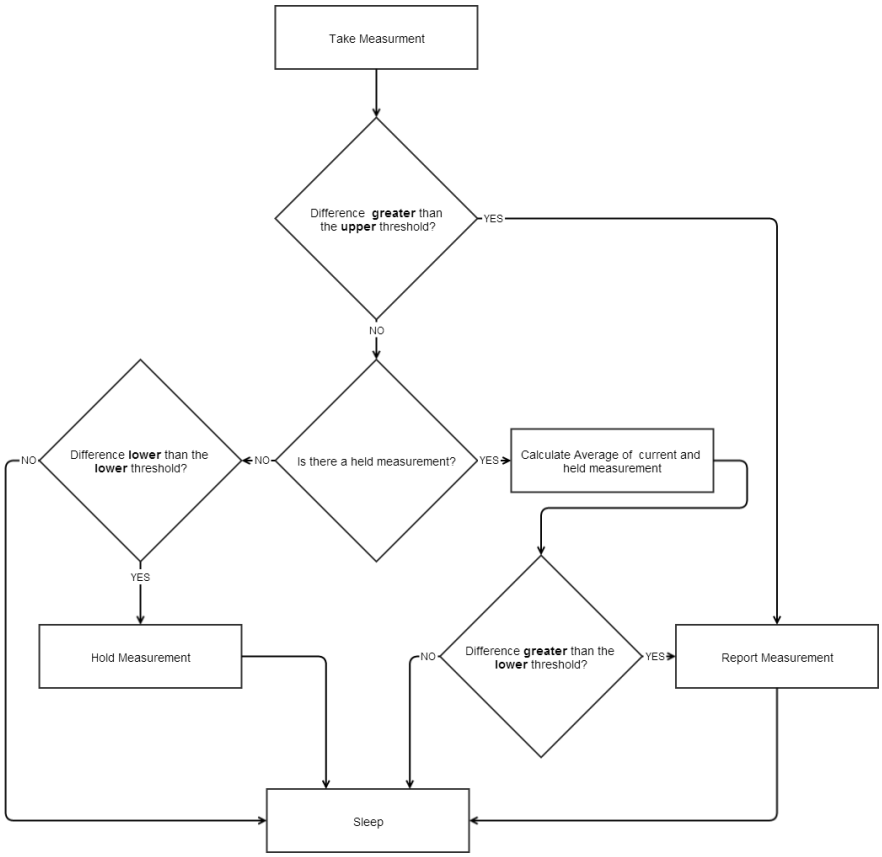


Figure 9.1: Flow chart of 2-tier measurement filtering.

Transmission Control Protocol (TCP) connection is established and a Hypertext Transfer Protocol (HTTP) request is transmitted over it. In case the UDP server is selected, a datagram is sent. Then, the firmware disconnects from the AP, turns the radio off and goes to sleep. All communication with the AP is encrypted through the Wi-Fi Protected Access II (WPA2) security protocol that is implemented in Wi-Fi. The initial association to the AP is performed using Wi-Fi Protected Setup (WPS).

9.2.3 Power Consumption and Charging Efficiency

For the experiments presented in this section a capacitor of 2.1 F is used for energy storage, instead of a lithium battery. Figure 9.2 shows the platform used for the exper-

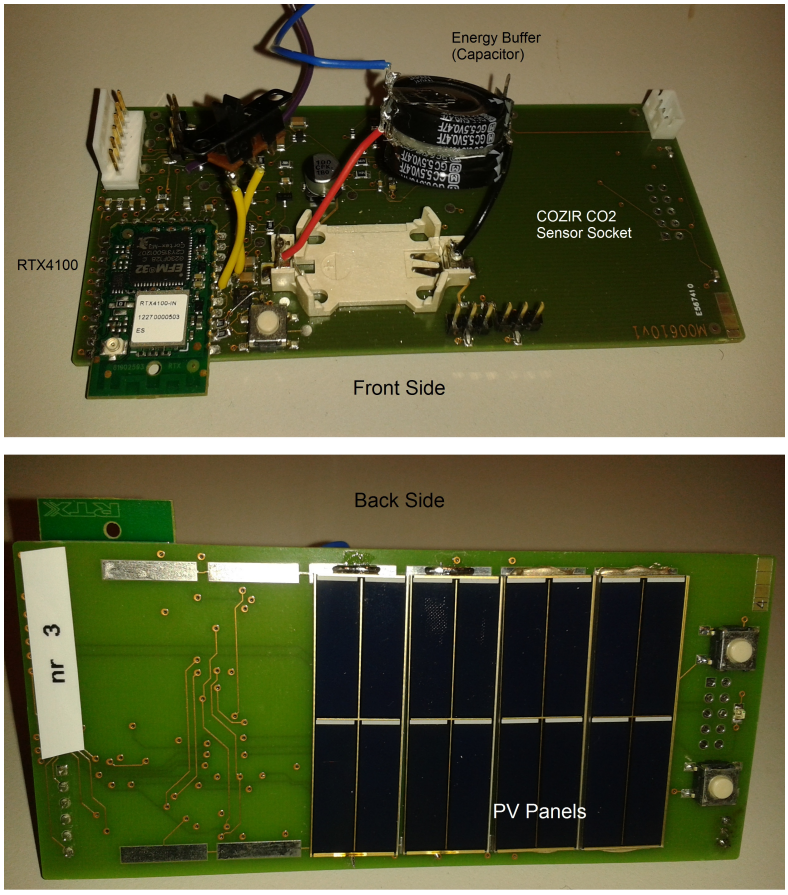


Figure 9.2: The prototype Energy Harvesting CO2 Sensor node.

iments.

With the transmission power set to 10 dBm, the current consumption while the radio is active peaks at approximately 120 mA. In case of UDP, the duration of the active period is approximately 2.5 seconds, which matches the results of the experiments presented in [101]. In case of HTTP, the duration of the active period varies between 2.5 and 5 seconds due to packet retransmissions by TCP. Figure 9.3 and Figure 9.4 show the current drain in a typical cycle measured across a 1 Ω shunt resistor, for UDP and HTTP respectively. Figure 9.5 shows the current drain when the CO2 sensor is activated, measured across a 10 Ω shunt resistor. After the initialization, the current periodically peaks at approximately 14 mA. The measurements verify that the radio communication dominates the energy consumption.

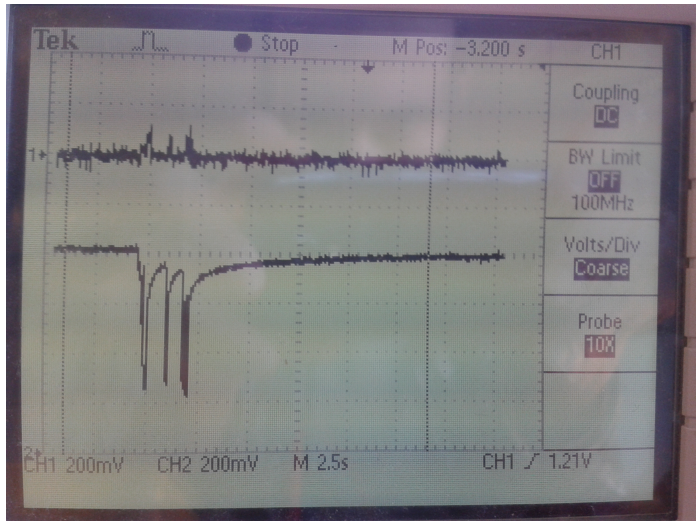


Figure 9.3: A typical duty cycle with UDP. The current drain can be estimated by dividing the voltage of the shunt resistor (upper line) over its resistance (1 Ω). The lower line shows the voltage of the storage capacitor.

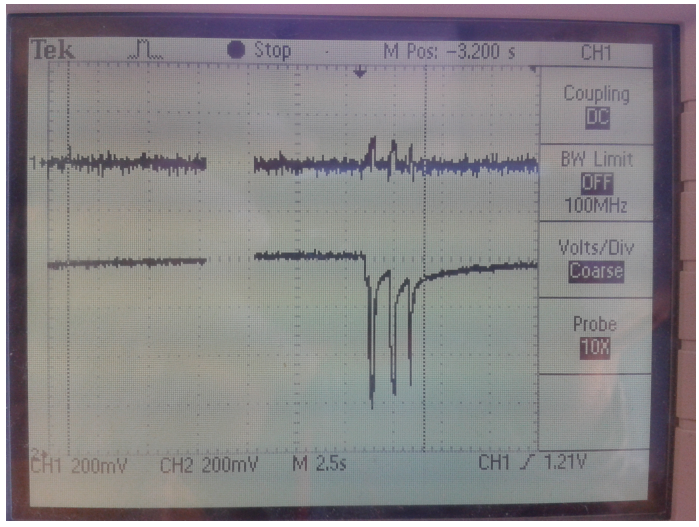


Figure 9.4: A typical duty cycle with HTTP. The current drain can be estimated by dividing the voltage of the shunt resistor (upper line) over its resistance (1 Ω). The lower line shows the voltage of the storage capacitor.

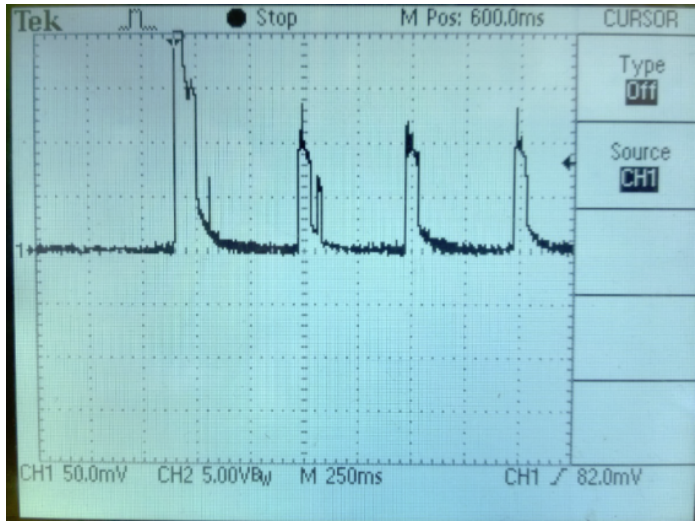


Figure 9.5: The activity of the CO2 sensor. The current drain can be estimated by dividing the voltage of the shunt resistor over its resistance (10 Ω).

The idle current drain was measured using two different methods. The first method is measuring the voltage of the $1\ \Omega$ shunt resistor while the sensor node is in idle mode. The constant current while sleeping is measured approximately $6\ \mu$A. To verify the instantaneous measurement in a longer period of time, the second method measures the discharge of the capacitor in a period of 30 minutes. During this period the voltage of the capacitor decreased by 5mV, which translates to a constant current of $5.83\ \mu$A or a constant power consumption of $23.5\ \mu$W in sleeping mode.

Then, the efficiency of the charging unit is evaluated. For the experiments, a light source was placed at different distances from the solar panels. The voltage across the solar panels and the input current (measured across a $1\ \Omega$ shunt resistor) are used to calculate the power input at the solar panels, i.e. before the charging unit. The system was let to charge the capacitor for 10 minutes. The difference of the voltage of the capacitor is used to calculate the actual charging power after the charging unit. Figure 9.6 shows the charging efficiency as the ratio of the charging power over the input power for different levels of constant input power. The results indicate an approximately 85% charging efficiency, that falls to approximately 75% when the input power is below 200μW.

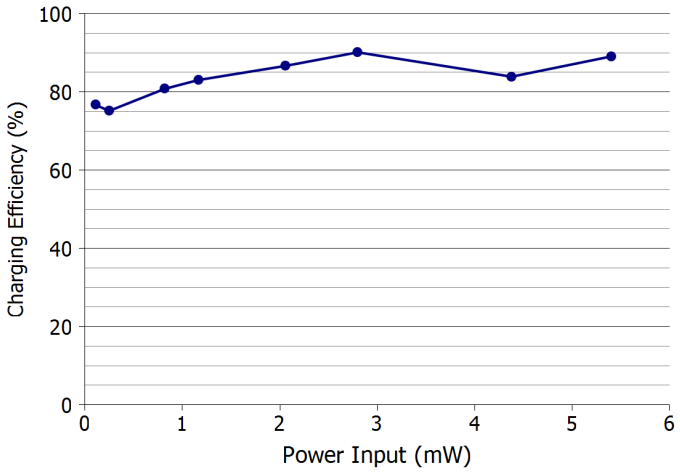


Figure 9.6: The efficiency of the charging unit, as the ratio of the charging power over the input power.

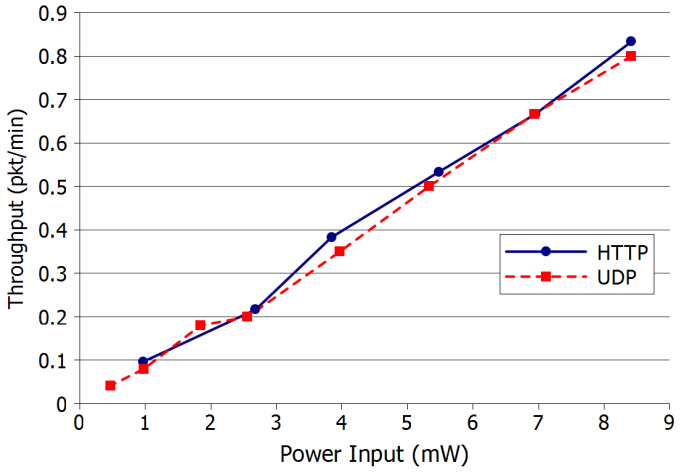


Figure 9.7: Sustainable performance at different levels of power input. The harvesting power density of ambient light, which depends heavily on the ambient excitation and harvesting technologies, is approximately 100 $\mu W/cm^2$ in an illuminated office and approximately 100 mW/cm^2 in direct sunlight [90].

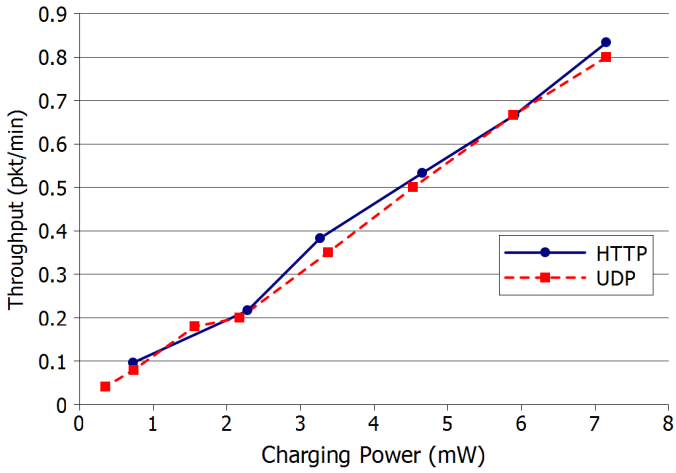


Figure 9.8: Sustainable performance at different levels of charging power.

9.2.4 Sustainable Operation

The following experiments aim to evaluate the sustainable performance of data transmission. The cost of using the CO2 sensor does not depend on the communication protocols used. Therefore, the CO2 sensor is deactivated and, instead, dummy data are transmitted to the server. The firmware is set to attempt one transmission every 30 second. The transmission is performed if and only if the voltage of the capacitor is above a threshold. This way, the system automatically finds balance and the sustainable throughput (in packets per minute) is measured. Again, the power input is controlled by positioning the light source in various distances from the solar cells. Figure 9.7 shows the results of the experiments for different levels of constant input power. All experiments were initiated with the voltage of the capacitor below the threshold. The 1 hour continuous operation demonstrates the sustainability of the node. Furthermore, the excess of harvested energy is used to improve the throughput of the application. The throughput increases linearly with the input power. HTTP and UDP seem to perform equally. This phenomenon is attributed to the power consumption of the association and the overhead protocols (DHCP and ARP) which is the same for both schemes and dominates the overall power consumption.

Figure 9.8 plots the results of the same experiment for different levels of charging power. The charging power is estimated using the charging efficiency that was measured in the previous experiment.

9.2.5 Comparison with ODMAC

The experiments shown in Figure 9.7 very closely resembles the experiments on ODMAC, shown in Figure 8.11. Both figures demonstrate a similar linear behavior where the throughput increases with the power input. Yet, ODMAC appears to require one order of magnitude less power for one order of magnitude more throughput. Moreover, as shown in Figure 8.8, most power consumed in ODMAC in idle listening in order to synchronize the sender to the duty-cycling receiver through the beacons. If the receiver did not have energy constraints, similarly to a Wi-Fi AP, the difference between the two protocols would be significantly higher.

Furthermore, data encryption has a significant effect on the energy-efficient of Wi-Fi sensor node. We experimentally verified the results shown in [101], which show that data encryption approximately doubles the energy consumption of the association. Similarly to ODMAC, the actual encryption of the data does not significantly increase the energy consumption of the packet transmission. However, the additional cost of the association to the AP, which occurs once every duty cycle, drives the overall energy consumption high and makes data encryption significantly less energy-efficient than ODMAC.

This comparison demonstrates that the benefits of using RTX4100 come at the price of compromising the energy-efficiency of the network. The two orders of magnitude of difference verify in practice that IEEE 802.11 and the TCP/IP stack are not energy-efficient solutions. Nevertheless, the use of IEEE 802.11 is feasible if the running application has loose performance requirements.

9.3 Timing Channels for Wireless Sensor Networks

A timing channel is a communication channel where the alphabet consists of different time values [85]. Galllager [44] was the first to study the information, packets can carry, beyond the information encoded in their payload. The work of Anantharam and Verdu [4], followed by the work of Bedekar and Azizoglu [9], identify the timing capacity of a channel and indicate that the overall capacity of a channel can be increased by encoding information in the interarrival times between packets originating from bursty sources. Communication through timing channels has been extensively studied in the literature, mainly from the perspective of system security [47, 52, 57, 85, 109]. Covert timing channels, that coexist along traditional data channels, constitute a means to secretly transmit information, which can be exploited by compromised systems to convey sensitive information without being detected.

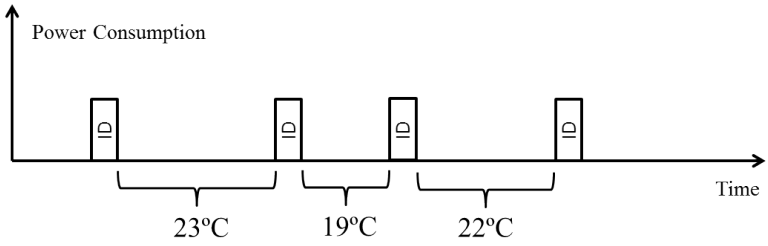


Figure 9.9: Motivational example of using timing channels in Wireless Sensor Networks. The temperature measurement information is encoded within the sleeping periods between control frame transmissions.

In WSNs, duty-cycling is a way to limit the useful operation of a node with the long-term goal of saving energy and extending its lifetime. In the context of WSNs, timing channels can be used to transmit information in a more energy-efficient manner. Instead of communicating in the traditional way [34], nodes can encode the measurement in the duration of their sleeping period, as shown in Figure 9.9. While using the radio is highly costly, modulating the sleeping time interval does not imply any energy consumption. While the measurement itself is, practically, transmitted with the radio off, there is still need for control data transmission in the traditional way. Nodes need to transmit a short frame and the measurement is encoded in the interarrival time between two sequential transmissions. Such a short frame needs to contain identification information, so that the receiver is able to identify sequential transmissions from the same source and calculate the interarrival time. This approach introduces a new perspective towards the realization of energy-efficient wireless networks.

To provide some intuition on this method, in this section, timing channels are modeled and analyzed with respect to the traditional data channel. The energy consumption improvements, that can be achieved, are identified, along with the effect of channel and timing errors.

9.3.1 Analytical Model

We consider a single-hop WSN (nodes form a star topology), in which sensor nodes report their measurements to a central data collector. We assume that the data collector does not have energy constraints and, therefore, has its radio continuously in receiving mode. A sensor node is identified with a unique number. Let k be the size of this identification number in bits. The size of a measurement in bits is defined as m and has a value $v \in [0, 2^m - 1]$. For simplicity, we consider discrete time. The duration of a timeslot is the time required to transmit a single bit. Therefore, a sensor node needs

$K = k$ timeslots to transmit its id and $M = m$ timeslots to transmit the measurement. Additionally, we define as c the energy per timeslot consumed for transmitting. During each timeslot a sensor node can either transmit a bit or sleep. In the latter case, the energy consumption is zero. Initially, we also assume that there are no channel errors and that the system does not introduce propagation and processing delays. Later in the analysis, we will remove these assumptions.

9.3.1.1 Model for the traditional data channel

The sensor node duty-cycles to save energy. Let S be the duration of sleeping between two consecutive transmissions in timeslots. The duty cycle (P) of a sensor node consists of a data transmission (id and measurement) that is followed by sleeping.

$$P = K + M + S \tag{9.1}$$

The energy consumed per transmission (E) is proportional to the size of the transmitted data.

$$E = c(K + M) \tag{9.2}$$

The throughput (T) in measurements per timeslot is estimated as follows.

$$T = \frac{1}{P} = \frac{1}{K + M + S} \tag{9.3}$$

The long-term average power consumption is C, where c is the power consumed for transmitting.

$$C = \frac{E}{T} = c\frac{K + M}{K + M + S} \tag{9.4}$$

The above equations model a duty-cycling node in the traditional case, where the throughput and the long-term average power consumption depend on the duration of sleep.

9.3.1.2 Model for the timing channel

Now consider that the node encodes the measurement information in the interarrival times between two transmissions. To do so, the node adds a sleeping delay of D timeslots to its duty cycle, such that $D = \theta v$, where v is the value of the measurement and $\theta \geq 1$ is a parameter that spreads the information across multiple timeslots. The duty cycle (P') consists of the id transmission that is followed by the sleeping period, S', and the sleeping delay associated with the encoded measurement (D).

$$P' = K + S' + D = K + S' + \theta v \tag{9.5}$$

Assuming that I is the interarrival time between two transmissions, the receiver is able to extract the value of the measurement using the following formula.

$$v = \frac{I - K - S'}{\theta} \tag{9.6}$$

Similarly to the traditional case, the energy consumed per transmission (E') is proportional to the size of the transmitted data.

$$E' = cK \tag{9.7}$$

The throughput (T') in measurements per timeslot is estimated as follows.

$$T' = \frac{1}{P'} = \frac{1}{K + S' + D} \tag{9.8}$$

The long-term average power consumption is C'.

$$C' = \frac{E'}{T'} = c\frac{K}{K + S' + D} \tag{9.9}$$

9.3.1.3 Gain and throughput constraints

By calculating the ratio of (9.2) over (9.7) we can estimate the energy savings of using the timing channel instead of the traditional channel to transmit the measurement.

$$g_E = \frac{E}{E'} = \frac{K+M}{K} = 1 + \frac{M}{K} \tag{9.10}$$

Moreover, the use of the timing channel introduces a constraint on the maximum throughput we can obtain from the system. This happens because the maximum throughput depends on the value of the measurement. The maximum throughput (T'_{max}) can be estimated by using the expected value of the measurement and assuming no additional sleeping time ($S' = 0$).

$$T'_{max} = \frac{1}{K + \theta E[v]} \tag{9.11}$$

This fact does not limit the throughput of the system if the intended sleeping time S, assuming the traditional case, satisfies equation (9.12) but constitutes a constraint otherwise.

$$S \geq \theta E[v] - M \tag{9.12}$$

Equations (9.10) and (9.11) indicate that by sacrificing the maximum achievable throughput, timing channels can significantly reduce the energy consumption per measurement. It should be noted that such a sacrifice is not different to what WSNs typically do. Being networks that primarily focus on energy-efficiency, the concept of duty cycles is, by definition, a tool that allows the network to sacrifice throughput for energy.

9.3.1.4 Timing errors in the timing channel

Continuing the analysis, let us now assume that the system inserts propagation and processing delays (N in timeslots). Such delay acts as noise, as it is added in the interarrival time (I) and affects the decoding of the measurement (9.6), $I = P' + N$. Note that such an error in the decoding of the measurement does not have the same catastrophic effects as channel errors. In fact, the measurement will be shifted to a close higher value that can or cannot be tolerated by the application. Furthermore, assuming

that the delay is constant, the error can be corrected by recalibrating the measurement, i.e. subtracting a constant number from the measured I.

Nevertheless, the θ parameter can be used to mitigate the error. By including N on the decoding formula (9.6) we notice that the error (e) depends on θ.

$$e = \frac{N}{\theta} \tag{9.13}$$

Essentially, the parameter θ is the equivalent of transmission power for the timing channel. With a sufficiently high value of θ, the error is eliminated.

$$\theta > N \tag{9.14}$$

Increasing θ mitigates or eliminates the errors, but also tightens the the maximum throughput constraint as presented in equation (9.11).

9.3.1.5 Channel errors in the timing channel

In the last part of the analysis, we introduce channel errors. Let us assume that there is a probability $p > 0$ that the data collector will not be able to decode the k bits of the identification number. The loss of an id frame can alter the interarrival time of the next measurement. Therefore, the receiver needs a way to guarantee that the interarrival time between two frames is not altered by a missing frame. An interarrival time is certainly altered by a missing frame if it is larger than the maximum acceptable value (I_{max}), obtained by (9.5) and considering $v = 2^m - 1$.

$$I_{max} = K + S' + \theta(2^m - 1) \tag{9.15}$$

To guarantee every case, we have to consider the worst case scenario, i.e. the interarrival time has its minimum value (I_{min}), obtained by (9.5) and considering $v = 0$. Errors due to channel errors, can be successfully identified if the following inequality is satisfied.

$$2I_{min} > I_{max} \Leftrightarrow S' > \theta(2^m - 1) - K \tag{9.16}$$

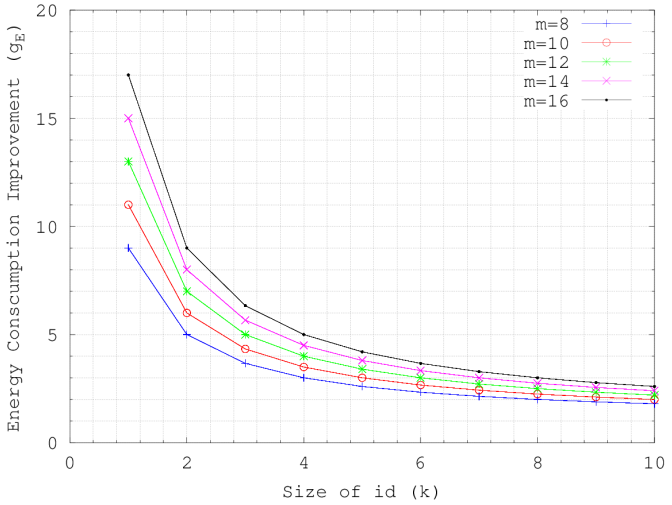


Figure 9.10: The energy consumption improvements for different sizes of the measurement (m) and the network (k).

Selecting a high enough S' to satisfy (9.16) makes error detection possible in the worst case scenario. Similarly to errors in the timing channel, channel errors can be handled by tightening the maximum throughput constraint.

9.3.2 Numerical Results

For the numerical results we assume using the Texas Instruments' eZ430 sensor nodes [115], which use the CC2500 radio [116]. CC2500 has a maximum transmission rate of $500\ Kbps$. Hence, the duration of each timeslot is $1.95\ \mu s$. The measurement, v, is obtained by an ADC with m bits of resolution, $m \in \{8, 10, 12, 14, 16\}$. For simplicity, we assume that the measurement is a random variable that follows a normal distribution and it is calibrated so its expected value is in the middle of the available space, $E[v] = 2^{m-1}$. Assuming that the propagation and processing delay is in the order of some milliseconds, we choose $\theta = 6000$, which satisfies (9.14) and translates to a minimum difference between two values of v of $11.72\ ms$.

Figure 9.10 demonstrates the improvements in energy consumption, as obtained from (9.10), for different values of m and k. We observe significant improvements that drop exponentially as the size of the network increases and increase linearly with the size of the measurement. Figure 9.11 shows the maximum throughput constraint in measurements per hour for different size of the measurement (m). The sleeping time

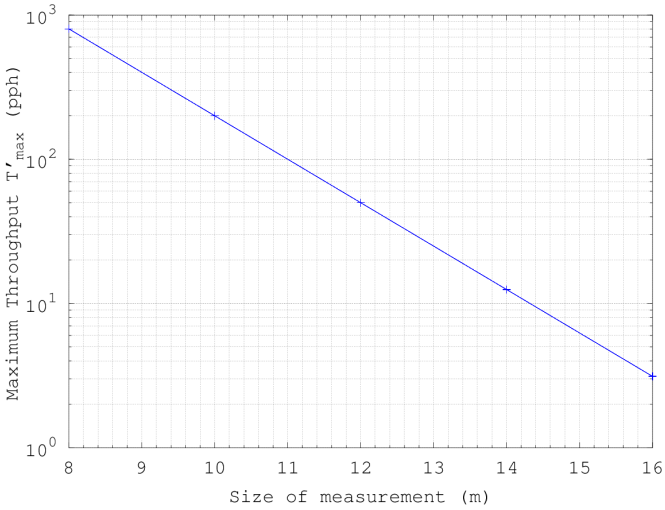


Figure 9.11: The maximum throughput constraint (in measurements per hour) considering the minimum acceptable sleeping time S' and for different sizes of the measurement (m).

(S') is set to the minimum value that satisfies (9.16). We observe that the maximum achievable throughput drops exponentially with the size of the measurement.

9.3.3 Discussion

Timing channels give rise to a new perspective towards the design of energy-efficient protocols for low-power WSNs. The analysis indicates that significant energy improvements can be achieved. Encoding the measurement in the interarrival times between control packets, forces the sleeping times to be sufficiently large to allow successful decoding that counters any unpredictable source of delay. Therefore, timing channels are suitable for low power applications that tolerate low throughput. As an extreme example, consider the application described in [121], where each node needs to operate with less than 10 μW of long-term average power consumption, but the application requires one measurement per day. With a sufficiently small measurement, the maximum throughput constraint can be tuned to be one measurement per some minutes, which does not constitute a constraint for many low-power sensor applications.

Beyond the promising initial results, timing channels introduce new questions and challenges, such as the scalability of the channel with regards to an increasing amount of nodes that attempt to use it. Intuitively, we expect channel collisions to be less than

in the traditional approach, as the nodes utilize the channel less. Yet, randomization techniques, that add a random delay between transmissions in an attempt to enforce random channel access (for example, see RI-MAC [107]) are not trivially applicable. In practice, such a random delay introduces noise in the timing channel and can be countered by adjusting the θ parameter. On the other hand, in wireless sensor networks that use timing channels, random channel access highly depends on the entropy of the measured variable. In other words, if the measured variable has sufficient spatial and temporal variations, timing channels will insert sufficient randomization in random channel access.

The capacity of the timing channel highly depends on the θ parameter and therefore the level of timing errors. Hence, identifying and countering unwanted sources of delay constitutes an interesting challenge. As mentioned in Section 9.3.1.4, upon being identified, any constant delay can be effectively used to correct the decoded value. In fact, sensor nodes are relatively simple computing systems, where typically there is just one thread running. Therefore, the processing delay is expected to be highly constant. On the other hand, the effect of clock drifts in the microprocessor should also be taken into consideration.

The current work also assumes that the whole information of the measurement is encoded in a single symbol and, therefore, does not scale well with the size of the measurement. An alternative approach would be to split the message in several symbols and transmit an additional control packet for every symbol. At the cost of decreasing the energy savings, this solution would allow larger messages and loose the maximum throughput constraints.

9.4 Summary

In this chapter, we are interested in links between duty-cycling senders and always-on receivers. In this context, two distinct works are presented.

In the first work, we present the development of a carbon dioxide sensor node that is powered by artificial light. The sensor node uses Wi-Fi for wireless communication, which is the protocol commonly used in wireless local area networks. We show experiments that demonstrate sustainable operation. The results are compared with ODMAC and indicate that radio communication with ODMAC is two orders of magnitude more energy-efficient.

Next, we discuss the idea of using timing channels in WSNs. Timing channels are communication channels in which the message is encoded into different time intervals. To promote energy-efficiency, the measurement can be encoded in the duration

of the sleeping time of a sensor node between the transmission of two sequential control frames. We provide a simple model of the timing channel that aims to estimate the potential improvement in the energy consumption. The numerical results indicate significant energy savings under realistic scenarios.

CHAPTER 10

Concluding Remarks

10.1 Overview

Concluding the dissertation, Section 10.2 discusses some open issues and proposes directions towards their solution, while Section 10.3 summarizes the contributions of the conducted research.

10.2 Discussion on Open Issues

Asynchronous receiver-initiated MAC protocols, including ODMAC, constitute the state-of-the-art approach for the establishment of links and the communication between senders and receivers that duty cycle. Receiver-initiated MAC protocols, in particular, need to face the challenge of the energy consumption overhead of finding a moment in time that both the sender and the receiver are active and available to exchange information. This challenge has led to significant research on the minimization of idle listening during the establishment of the link. The unpredictable and ever-changing nature of energy harvesting constitutes many of the approaches against idle listening, such as the prediction of predefined wake-up times, inapplicable. We proposed opportunistic forwarding as a means to significantly decrease idle listening. Opportunistic forwarding

dictates that a sender has multiple forwarding options and every time uses the one that is available first.

This approach requires the routing layer to provide the MAC layer with a list of forwarding candidates, which are chosen with respect to routing metric. The cross-layer optimization between the MAC layer and routing layer is vital for the efficient use of the energy resources of a sensor node. The presented research on the MAC layer suggests that idle listening for the establishment of the link is the dominant source of energy consumption. Therefore, the routing layer should route the packets through paths that minimize the idle listening. While the investigation of a routing protocol that cooperates efficiently with ODMAC is an open issue, the insight obtained from the presented results leads to expectation that the simplest routing metric that minimizes the hops to the sink node (for instance LAR 3.6.4), will lead to the most energy-efficient paths.

A different challenge, that is exposed by the presented results, is that energy harvesting may potentially lead to topologies that are imbalanced with regards to energy resources. In fact, multi-hop wireless networks are imbalanced even when they are battery-powered. The closer a sensor node is to the sink node, the more forwarding duties it has. In addition to that, spatial variations in energy harvesting can make matters worse. Specifically, it may lead to scenarios that receivers with low energy resources are flooded by many data packets from nodes that are more energy rich.

We propose the exploitation of the AB mechanism as a means to avoid flooded situations in an fully distributed and energy-efficient manner. In particular, in a flooded situation, the receivers will be able to provide only few beacons to the senders that they serve. Therefore, the senders will see the need to back off more frequently. A high frequency of backoff events is a clear indication that the receivers are asked to serve more packets than what they are capable of. Senders can react to this indication in an attempt to balance the network. By decreasing their sensing frequency, senders will directly contribute against the flood, as they will inject the network with less packets. Moreover, by decreasing their beaconing frequency, they will contribute against the flood in two indirect ways. The nodes of the next layer will, first, tend to choose alternative paths to the sink, due to opportunistic forwarding. If the beacons are not enough, they will see an increase in the frequency of their backoff events and they will decrease their own duty cycles. As a result, the need to react to the flood will propagate, up to the outer layer of the network, until the situation is resolved. The validation and evaluation of this mechanism, as well as its incorporation to ODMAC, remains an open topic.

With respect to timing channels in links with receivers that are always active, Section 9.3.3 summarizes several directions for future investigations. In addition to those, cross-layer optimization is also important. Specifically, the overall system-wide energy consumption improvements of encoding the measurement in the duration of the sleeping period, heavily depends on the energy consumption of the other layers and specifically the overhead of the physical layer. The physical layer, typically, uses a

preamble bit sequence that aims to synchronize the demodulator of the receiver to the received signal. To fully investigate the potential gains of using timing channels, the system should use radios with demodulators that minimize the size of the preamble or, ideally, support preamble-less synchronization techniques (e.g. [31]).

10.3 Conclusion

With a primary interest on the links that both the receivers and the senders are alternating between active and sleeping states to save energy, this dissertation focuses on the receiver-initiated paradigm of asynchronous communication. The receiver-initiated paradigm synchronizes a duty-cycling receiver to a duty-cycling sender with beacons. Whenever the receiver is ready, it transmits a beacon that indicates his availability to receive data. The sender, silently listens the channel, waiting for a beacon from the intended receiver. MAC protocols that follow the receiver-initiated paradigm incorporate features that deal with several challenges of the link layer, including collision avoidance, idle listening mitigation and provision of QoS. The presented survey of all the receiver-initiated MAC protocols (Chapter 2) summarizes the particular features that each protocol offers and discusses on which features fit best under different environmental conditions and system constraints. Furthermore, we stressed that features from different protocols can be selected and combined into new protocols that are tailored for specific needs.

Focusing on sensor networks that are powered by energy harvesting (i.e. EH-WSNs), we presented a receiver-initiated MAC protocol, named ODMAC, which incorporates and investigates several unique optimization features (Chapter 3). These features are tools for a network designer that aim to address the challenges of idle listening, collision avoidance, adaptivity, security and QoS. The key features of ODMAC are: (i) adaptive duty cycles, (ii) opportunistic forwarding, (iii) collision avoidance and traffic differentiation with AB and (iv) RAP.

ODMAC autonomously adapts the duty cycles of sensing and forwarding (i.e. beaconing) to balance the energy consumption to the available harvested energy in order to provide sustainable operation. Aiming to support the sustainability of the network and the application performance, ODMAC incorporates opportunistic forwarding, a forwarding mechanism that dictates that a sender has multiple forwarding alternatives and uses the best one in a per-packet basis. The performance of opportunistic forwarding and adaptive duty cycles is evaluated through analysis and simulations in OPNET (Chapter 4). The results from both sources indicate that nodes are able to achieve sustainable operation in various realistic energy conditions. At the same time, any excess of energy is used to favor different application-specific priorities, such as delay and throughput. With respect to opportunistic forwarding, the presented analysis verifies

that the feature significantly reduces the energy consumed in idle listening and promotes the autonomous load balancing of the forwarding duties to the sensor nodes that have access to more energy.

AB is a collision avoidance mechanism that aims to avoid inevitable collisions before the beacon transmission, so that the contending senders can back off without consuming energy in idle listening. The performance of AB is evaluated and compared to the state-of-the-art collision avoidance mechanism in wireless networks (Chapter 5). Simulation results indicate that AB improves the energy-efficiency of the network. Furthermore, the results suggest that AB is long-term fair, i.e. provides equal opportunities for channel access, and scales well with increasing levels of contention. Furthermore, AB is able to provide QoS by prioritizing urgent traffic, such as alerts, and sacrificing less important data packets.

Network intruders can trivially capture beacons and replay them while their creator is in a sleeping state. Beacon replay attacks constitute building blocks for DoS attacks. RAP is a security extension of ODMAC that protects the network from such attacks. RAP is a challenge-response scheme that aims to authenticate the receiver in a receiver-initiated communication. The effectiveness of RAP against beacon replay attacks is validated using various verification tools (Chapter 6). Furthermore, analytical results highlight its energy-efficient nature and demonstrate the trade-off between the level of security, measured by the resilience of the scheme to space exhaustion, and the level of energy consumption.

The performance of ODMAC is also compared with two state-of-the-art MAC protocols that are widely used in either the academic or the industrial world (Chapter 7). The analytical comparison of the receiver-initiated ODMAC with an adaptive variation of the sender-initiated X-MAC demonstrates that ODMAC can be tuned to consume less energy. Therefore, it is more suitable when the available environmental energy is low and when the application requires the system to operate at a duty cycle that minimizes energy consumption. The analytical comparison of ODMAC to the industrial protocol IMR+, which is currently used in a large-scale commercial network, demonstrates that ODMAC is more suitable for energy harvesting applications, as it is able to dynamically manage the energy resources to improve the performance of the application.

To strengthen the confidence of the analytical and simulation results, we provide a prototype implementation of ODMAC for Texas Instruments' eZ430-rf2500 wireless sensor nodes (Chapter 8). The conducted testbed experiments demonstrate sustainable links that use the available harvested energy to favor different application performance metrics, including throughput and link delay. Moreover, the experiments verify that AB is effectively avoiding collisions in an energy-efficient manner, provides contending nodes with equal opportunities to access the channel and is able to prioritize urgent traffic.

The presented analysis, simulations and testbed experiments demonstrate that ODMAC effectively contributes to the fundamental system goals of EH-WSNs, namely long-term sustainability and energy-efficient application performance.

In the last part of the dissertation, our interest moves to links that only the sender duty-cycles while the receiver does not have any energy constraints and, therefore, is always in an active state. In this context, we present the development of a prototype energy-harvesting CO2 sensor node that operates with IEEE 802.11 [55]. The key advantage of developing sensing applications with this approach, is the compatibility with existing networks and infrastructures. The experiments demonstrate sustainable operation for different levels of power input. Furthermore, the experimental results indicate that the advantages of using IEEE 802.11 comes at the price of compromising the energy-efficiency of the node. Even without the overhead of synchronizing duty-cycling nodes, ODMAC is two orders of magnitude more energy-efficient.

Lastly, we discuss the idea of using timing channels to promote the energy-efficiency of WSNs. Instead of conveying information in the traditional way, senders can encode the measurement into the duration of the sleeping period. Initial analytical results suggest substantial reduction of the energy consumption under realistic scenarios and motivate future investigations.

The MAC layer plays a critical role towards the realization of low-power sensor applications and leads the research community to push the envelope towards increasing the energy-efficiency of wireless communications. In order to meet tight energy constraints, sensing systems need to be optimized as a whole and tailored to the specific environmental conditions of each given application. As there is no globally optimal solution, researchers provide the designers of WSNs with tools and features that can be adapted and used with respect to particular application requirements. It is the belief of the author that this dissertation provides significant insight and valuable tools that can be selected, altered or combined with other tools and contribute towards the realization of long-living and energy-efficient wireless sensing infrastructures.

Bibliography

[1] Skipjack and kea algorithm specifications. Technical report, May 1998.

[2] Norman Abramson. THE ALOHA SYSTEM - Another alternative for computer communications. In *Proc. Fall Joint Comput. Conf.*, pages 281–285. ACM, 1970.

[3] Kemal Akkaya and Mohamed Younis. A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks*, 3:325–349, 2005.

[4] V. Anantharam and S. Verdu. Bits through queues. *IEEE Trans. on Inform. Theory*, 42(1):4–18, 1996.

[5] Th. Arampatzis, J. Lygeros, and S. Manesis. A Survey of Applications of Wireless Sensors and Wireless Sensor Networks. In *Proc. IEEE Mediterrean Conf. on Control and Automation*, pages 719–724, 2005.

[6] AVISPA. Deliverable 2.3: The Intermediate Format, 2003. Available at `http://www.avispa-project.org`.

[7] A. Bachir, M. Dohler, T. Watteyne, and K.K. Leung. MAC Essentials for Wireless Sensor Networks. *IEEE Commun. Surveys Tutorials*, 12(2):222–248, 2010.

[8] D. Basin, S. Mödersheim, and L. Viganò. OFMC: A symbolic model checker for security protocols. *Int. J. of Inform. Security*, 4(3):181–208, 2005.

[9] A.S. Bedekar and M. Azizoglu. The information-theoretic capacity of discrete-time queues. *IEEE Trans. on Inform. Theory*, 44(2):446–461, 1998.

[10] Bruno Blanchet. An Efficient Cryptographic Protocol Verifier Based on Prolog Rules. In *Proc. 14th IEEE Comput. Security Foundations Workshop (CSFW-14)*,

pages 82–96, Cape Breton, Nova Scotia, Canada, June 2001. IEEE Computer Society.

[11] Bob Fornaro, Mick Kulikowski and Kathleen Angione. Tiny Sensor-Based Computers Could Help Track Wildlife. North Carolina State University, 2003.

[12] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. Robshaw, Y. Seurin, and C. Vikkelsoe. PRESENT: An Ultra-Lightweight Block Cipher. In *Proc. 9th Int. Workshop on Cryptographic Hardware and Embedded Syst. (CHES)*, pages 450–466, 2007.

[13] Michael Buettner, Gary V. Yee, Eric Anderson, and Richard Han. X-MAC: A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks. In *Proc. 4th ACM Int. Conf. on Embedded Networked Sensor Syst. (SenSys)*, pages 307–320. ACM, 2006.

[14] C. Cano, B. Bellalta, A. Sfairopoulou, and M. Oliver. Low energy operation in WSNs: A survey of preamble sampling MAC protocols. *Computer Networks*, 55(15):3351–3363, 2011.

[15] CEN/TC 294. Communication systems for meters and remote reading of meters - Part 4: Wireless meter readout (Radio meter reading for operation in the 868 MHz to 870 MHz SRD band), 2005.

[16] S. Chalasani and J.M. Conrad. A survey of energy harvesting sources for embedded systems. In *IEEE Southeastcon*, pages 442–447, 2008.

[17] CO2 Meter. COZIR: Ultra Low Power Carbon Dioxide Sensor. `http://www.co2meters.com/Documentation/Datasheets/DS-COZIR-Ambient-CO2.pdf`.

[18] Cymbet Corporation. EnerChip CC Energy Harvester Evaluation Kit. CBC-EVAL-10, DS-72-20 Rev A, 2011. Available at `http://www.cymbet.com/pdfs/DS-72-20.pdf`.

[19] Cymbet Corporation. EnerChip EP Universal Energy Harvester Eval Kit. CBC-EVAL-09, DS-72-13 Rev E, 2012. Available at `http://www.cymbet.com/pdfs/DS-72-13.pdf`.

[20] Cymbet Corporation. Using the EnerChip in Pulse Current Applications. AN-1025, 2012. Available at `http://www.cymbet.com/pdfs/AN-1025.pdf`.

[21] Tijs Van Dam and Koen Langendoen. An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks. In *Proc. 1st ACM Int. Conf. on Embedded Networked Sensor Syst. (SenSys)*, pages 171–180. ACM, 2003.

[22] David Pescovitz. Brainy Buildings Conserve Energy. UC Berkeley College of Engineering Lab Notes, 2001.

[23] Jing Deng, Richard Han, and Shivakant Mishra. Limiting DoS attacks during multihop data delivery in wireless sensor networks. *Int. J. Secur. Netw.*, 1(3/4), 2006.

[24] Dorothy E. Denning and Giovanni Maria Sacco. Timestamps in key distribution protocols. *Commun. ACM*, 24(8):533–536, 1981.

[25] Alessio Di Mauro, Xenofon Fafoutis, Sebastian Mödersheim, and Nicola Dragoni. Detecting and Preventing Beacon Replay Attacks in Receiver-Initiated MAC Protocols for Energy Efficient WSNs. In *Proc. of the 18th Nordic Conf. on Secure IT Syst. (NordSec)*, volume 8208, pages 1–16, 2013.

[26] Alessio Di Mauro, Davide Papini, and Nicola Dragoni. Security Challenges for Energy-harvesting Wireless Sensor Networks. In *Proc. 2nd Int. Conf. on Pervasive Embedded Computing and Commun. Syst.*, pages 422–425, 2012.

[27] D. Dolev and A. Yao. On the security of public key protocols. *IEEE Trans. Inform. Theory*, 29(2):198–208, 2006.

[28] Jing Dong, Kurt E. Ackermann, Brett Bavar, and Cristina Nita-Rotaru. Mitigating attacks against virtual coordinate based routing in wireless sensor networks. In *Proc. 1st ACM Conf. on Wireless Network Security*, pages 89–99. ACM, 2008.

[29] A. Dunkels, B. Gronvall, and T. Voigt. Contiki - a lightweight and flexible operating system for tiny networked sensors. In *Proc. 29th IEEE Int. Conf. on Local Comput. Networks*, pages 455–462, 2004.

[30] Prabal Dutta, Stephen Dawson-Haggerty, Yin Chen, Chieh-Jan Mike Liang, and Andreas Terzis. A-MAC: A versatile and efficient receiver-initiated link layer for low-power wireless. *ACM Trans. Sensor Networks*, 8(4):30:1–30:29, September 2012.

[31] D. Efstathiou and A.H. Aghvami. Preamble-less nondecision-aided (NDA) feedforward synchronization techniques for 16-QAM TDMA demodulators. *IEEE Trans. Vehicular Technology*, 47(2):673–685, 1998.

[32] Amre El-Hoiydi and Jean-Dominique Decotignie. WiseMAC: An Ultra Low Power MAC Protocol for Multi-hop Wireless Sensor Networks. In *Proc. 1st Int. Workshop on Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS)*, pages 18–31, 2004.

[33] Energy Micro. EFM32 Gecko 32-bit Microcontroller. `http://www.silabs.com/products/mcu/lowpower/pages/efm32g-gecko.aspx`.

[34] Zhi Ang Eu, Winston K. G. Seah, and Hwee-Pink Tan. A Study of MAC Schemes for Wireless Sensor Networks Powered by Ambient Energy Harvesting. In *Proc. 4th Int. Conf. on Wireless Internet (WICON)*, pages 78:1–78:9, 2008.

[35] Zhi Ang Eu, H. Tan, and W.K.-G. Seah. Routing and Relay Node Placement in Wireless Sensor Networks Powered by Ambient Energy Harvesting. In *Proc. IEEE Wireless Commun. and Networking Conf. (WCNC)*, pages 1–6, 2009.

[36] Xenofon Fafoutis, Alessio Di Mauro, and Nicola Dragoni. Sustainable Medium Access Control: Implementation and Evaluation of ODMAC. In *Proc. IEEE Int. Conf. on Commun. (ICC) Workshops*. IEEE, 2013.

[37] Xenofon Fafoutis, Alessio Di Mauro, and Nicola Dragoni. Sustainable Performance in Energy Harvesting - Wireless Sensor Networks. In *Proc. 4th ACM Int. Conf. on Future Energy Systems (e-Energy)*. ACM, 2013.

[38] Xenofon Fafoutis and Nicola Dragoni. ODMAC: An On-Demand MAC Protocol for Energy Harvesting - Wireless Sensor Networks. In *Proc. 8th ACM Symp. on Performance Evaluation of Wireless Ad-Hoc, Sensor, and Ubiquitous Networks (PE-WASUN)*, pages 49–56. ACM, 2011.

[39] Xenofon Fafoutis and Nicola Dragoni. Adaptive media access control for Energy Harvesting - Wireless Sensor Networks. In *Proc. 9th Int. Conf. Networked Sensing Syst. (INSS)*, pages 1–4, 2012.

[40] Xenofon Fafoutis and Nicola Dragoni. Analytical Comparison of MAC Schemes for Energy Harvesting-Wireless Sensor Networks. In *Proc. 9th Int. Conf. Networked Sensing Syst. (INSS)*. IEEE, 2012.

[41] Xenofon Fafoutis, Charalmpos Orfanidis, and Nicola Dragoni. Altruistic Backoff: Collision Avoidance for Receiver-Initiated MAC Protocols for Wireless Sensor Networks. *Int. J. of Distributed Sensor Networks*, 2013.

[42] Xenofon Fafoutis, Thomas Søresen, and Jan Madsen. Energy Harvesting - Wireless Sensor Networks for Indoors Applications using IEEE 802.11. In *Proc. 5th International Conference on Ambient Systems, Networks and Technologies (ANT), Procedia Computer Science Vol. 32C*, pages 1002–1007. Elsevier, 2014.

[43] Xenofon Fafoutis, Madava D. Vithanage, Alessio Di Mauro, and Nicola Dragoni. Receiver-Initiated Medium Access Control Protocols for Wireless Sensor Networks. Manuscript submitted for publication. *Comput. Networks J.*, 2013.

[44] Robert Gallager. Basic limits on protocol information in data communication networks. *IEEE Trans. on Inform. Theory*, 22(4):385–398, 1976.

[45] Amrita Ghosal, Subir Halder, Sanjib Sur, Avishek Dan, and Sipra DasBit. Ensuring basic security and preventing replay attack in a query processing application domain in WSN. In *Proc. Int. Conf. on Computational Science and its Applications*, pages 321–335. Springer-Verlag, 2010.

[46] James Gilbert and Farooq Balouchi. Comparison of energy harvesting systems for wireless sensor networks. *Int. J. of Automation and Computing*, 5(4):334–347, 2008.

[47] J. Giles and B. Hajek. An information-theoretic and game-theoretic study of timing channels. *IEEE Trans. on Inform. Theory*, 48(9):2455–2477, 2002.

[48] G. P. Halkes, T. Van Dam, and K. G. Langendoen. Comparing Energy-Saving MAC Protocols for Wireless Sensor Networks. *Mobile Networks and Applications*, 10(5):783–791, 2005.

[49] W.B. Heinzelman, A.P. Chandrakasan, and H. Balakrishnan. An application-specific protocol architecture for wireless microsensor networks. *IEEE Trans. on Wireless Commun.*, 1(4):660–670, 2002.

[50] Alfred Horn. On Sentences Which are True of Direct Unions of Algebras. *J. Symb. Log.*, pages 14–21, 1951.

[51] Qian Hu, Qiming Tian, and Zhenzhou Tang. RP-MAC: A Passive MAC Protocol with Frame Reordering for Wireless Sensor Networks. *Int. J. of Wireless Inform. Networks*, 20(1):74–80, 2013.

[52] Wei-Ming Hu. Reducing timing channels with fuzzy time. *J. of Comput. Security*, 1(3):233–254, 1992.

[53] Pei Huang, Chen Wang, Li Xiao, and Hongyang Chen. RC-MAC: A Receiver-Centric Medium Access Control Protocol for Wireless Sensor Networks. In *Proc. 18th Int. Workshop on Quality of Service (IWQoS)*, pages 1–9. IEEE, 2010.

[54] IEEE. IEEE Std. 802.15.4-2003: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (LR-WSNs). 2003.

[55] IEEE. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. 2012.

[56] M.K. Jakobsen, J. Madsen, and M.R. Hansen. DEHAR: A distributed energy harvesting aware routing algorithm for ad-hoc multi-hop wireless sensor networks. In *Proc. IEEE Int. Symp. on World of Wireless Mobile and Multimedia Networks (WoWMoM)*, pages 1–9, 2010.

[57] M.H. Kang, I.S. Moskowitz, and D.C. Lee. A network pump. *IEEE Trans. on Software Eng.*, 22(5):329–338, 1996.

[58] Aman Kansal, Jason Hsu, Sadaf Zahedi, and Mani B. Srivastava. Power Management in Energy Harvesting Sensor Networks. *ACM Trans. on Embedded Computing Syst. (TECS)*, 6(4):32, 2007.

[59] C. Karlof and D. Wagner. Secure routing in wireless sensor networks: attacks and countermeasures. In *Proc. 1st IEEE Int. Workshop on Sensor Network Protocols and Applications*, pages 113–127, 2003.

[60] Chris Karlof, Naveen Sastry, and David Wagner. TinySec: a link layer security architecture for wireless sensor networks. In *Proc. 2nd ACM Int. Conf. on Embedded Networked Sensor Syst. (SenSys)*, pages 162–175. ACM, 2004.

[61] SeongCheol Kim, JunHeon Jeon, and HyunJoo Park. QoS Aware Energy-Efficient (QAEE) MAC Protocol for Energy Harvesting Wireless Sensor Networks. In *Convergence and Hybrid Inform. Technology*, volume 7425, pages 41–48. Springer Berlin Heidelberg, 2012.

[62] T. Knot. Smart surrogates. *BP Frontiers Magazine*, 9:6–10, 2004.

[63] Daichi Kominami, Masashi Sugano, Masayuki Murata, and Takaaki Hatauchi. Energy-Efficient Receiver-Driven Wireless Mesh Sensor Networks. *MDPI Sensors*, 11(1):111–137, 2010.

[64] M. Kubisch, H. Karl, A. Wolisz, L.C. Zhong, and J. Rabaey. Distributed algorithms for transmission power control in wireless sensor networks. In *Proc. IEEE Int. Conf. on Wireless Commun. and Networking (WCNC)*, volume 1, pages 558–563, 2003.

[65] Q. Lampin, D. Barthel, I. Auge-Blum, and F. Valois. SARI-MAC: The Self Adapting Receiver Initiated MAC protocol for Wireless Sensor Networks. In *Proc. 8th IEEE Int. Conf. on Wireless and Mobile Computing, Networking and Commun. (WiMob)*, pages 12–18. IEEE, 2012.

[66] Emanuele Lattanzi, Edoardo Regini, Andrea Acquaviva, and Alessandro Bogliolo. Energetic Sustainability of Routing Algorithms for Energy-harvesting Wireless Sensor Networks. *Comput. Commun.*, 30(14-15):2976–2986, 2007.

[67] Hanjin Lee, Jaeyoung Hong, Suho Yang, Ingook Jang, and Hyunsoo Yoon. A Pseudo-Random Asynchronous Duty Cycle MAC Protocol in Wireless Sensor Networks. *IEEE Commun. Lett.*, 14(2):136–138, 2010.

[68] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler. TinyOS: An Operating System for Sensor Networks. In *Ambient Intelligence*, pages 115–148. Springer Berlin Heidelberg, 2005.

[69] Jinbao Li, Desheng Zhang, and Longjiang Guo. DCM: A Duty Cycle Based Multi-channel MAC Protocol for Wireless Sensor Networks. In *IET Int. Conf. on Wireless Sensor Network (IET-WSN)*, pages 233–238, 2010.

[70] Jinbao Li, Desheng Zhang, Longjiang Guo, Shouling Ji, and Yingshu Li. ARM: an Asynchronous Receiver-initiated Multichannel MAC Protocol with Duty Cycling for WSNs. In *Proc. 29th IEEE Int. Performance Computing and Commun. Conf. (IPCCC)*, pages 114–121. IEEE, 2010.

[71] En-Yi A. Lin, Jan M. Rabaey, and Adam Wolisz. Power-efficient rendez-vous schemes for dense wireless sensor networks. In *Proc. IEEE Int. Conf. on Commun. (ICC)*, volume 7, pages 3769–3776. IEEE, 2004.

[72] Longbi Lin, Ness B. Shroff, and R. Srikant. Asymptotically Optimal Energy-aware Routing for Multihop Wireless Networks with Renewable Energy Sources. *IEEE/ACM Trans. Netw.*, 15(5):1021–1034, 2007.

[73] Peng Lin, Chunming Qiao, and Xin Wang. Medium Access Control With A Dynamic Duty Cycle For Sensor Networks. In *Proc. IEEE Wireless Commun. and Networking Conf. (WCNC)*, volume 3, pages 1534–1539. IEEE, 2004.

[74] Shan Lin, Jingbin Zhang, Gang Zhou, Lin Gu, John A. Stankovic, and Tian He. ATPC: Adaptive Transmission Power Control for Wireless Sensor Networks. In *Proc. 4th Int. Con. on Embedded Networked Sensor Syst. (SenSys)*, pages 223–236. ACM, 2006.

[75] D. Liu and P. Ning. Efficient Distribution of Key Chain Commitments for Broadcast Authentication in Distributed Sensor Networks. Technical report, 2002.

[76] Gavin Lowe. A Hierarchy of Authentication Specifications. In *Proc. 10th Comput. Security Foundations Workshop (CSFW)*, pages 31–43. IEEE Computer Society Press, 1997.

[77] Bloomberg L.P. Tech Wave 2: The Sensor Revolution. *Bloomberg Business Week Magazine*, 2003.

[78] Alan Mainwaring, David Culler, Joseph Polastre, Robert Szewczyk, and John Anderson. Wireless sensor networks for habitat monitoring. In *Proc. of the 1st ACM Int. Workshop on Wireless Sensor Networks and Applications (WSNA)*, pages 88–97. ACM, 2002.

[79] Mathworks, Inc. MATLAB: The Language of Technical Computing.

[80] Ueli M. Maurer and Pierre E. Schmid. A Calculus for Security Bootstrapping in Distributed Systems. *J. Comp. Security*, 4(1):55–80, 1996.

[81] Micropelt GmbH. TE-CORE7 ThermoHarvesting Power Module, 2012. Available at `http://www.micropelt.com/down/datasheet_te_core.pdf`.

[82] Sebastian Mödersheim. Algebraic Properties in Alice and Bob Notation. In *Int. Conf. on Availability, Reliability and Security (ARES)*, pages 433–440, 2009.

[83] Sebastian Mödersheim. Abstraction by set-membership: verifying security protocols and web services with databases. In *ACM Conf. on Comput. and Commun. Security*, pages 351–360, 2010.

[84] Sebastian Mödersheim and Luca Viganò. Secure Pseudonymous Channels. In *Proc. of Esorics*, pages 337–354. Springer-Verlag, 2009.

[85] Ira S Moskowitz and Allen R Miller. Simple timing channels. In *Proc. IEEE Comp. Society Symp. on Research in Security and Privacy*, pages 56–64. IEEE, 1994.

[86] Razvan Musaloiu-E., Chieh-Jan Mike Liang, and Andreas Terzis. Koala: Ultra-Low Power Data Retrieval in Wireless Sensor Networks. In *Proc. of the 7th Int. Conf. on Inform. Process. in Sensor Networks*, pages 421–432. IEEE Computer Society, 2008.

[87] D. Niyato, E. Hossain, and A. Fallahi. Sleep and Wakeup Strategies in Solar-Powered Wireless Sensor/Mesh Networks: Performance Analysis and Optimization. *IEEE Trans. on Mobile Computing*, 6(2):221–236, 2007.

[88] OPNET Technologies, Inc. The OPNET Simulator.

[89] Amitangshu Pal. Localization Algorithms in Wireless Sensor Networks: Current Approaches and Future Challenges. *Network Protocols and Algorithms*, 2(1), 2010.

[90] J.A. Paradiso and T. Starner. Energy scavenging for mobile and wireless electronics. *Pervasive Computing, IEEE*, 4(1):18–27, 2005.

[91] Chulsung Park, K. Lahiri, and A. Raghunathan. Battery discharge characteristics of wireless sensor nodes: an experimental analysis. In *Proc. 2nd Ann. IEEE Commun. Soc. Conf. on Sensor, Mesh and Ad Hoc Commun. and Networks (SECON)*, pages 430–440. IEEE, 2005.

[92] Yang Peng, Zi Li, Daji Qiao, and Wensheng Zhang. Delay-Bounded MAC with Minimal Idle Listening for Sensor Networks. In *Proc. 30th Ann. Joint Conf. IEEE Comput. and Commun. Soc. (INFOCOM)*, pages 1314–1322. IEEE, 2011.

[93] J.K. Perng, Brian Fisher, S. Hollar, and K.S.J. Pister. Acceleration sensing glove (ASG). In *Proc. 3rd Intl. Symp. on Wearable Computers*, pages 178–180, 1999.

[94] Adrian Perrig, Robert Szewczyk, J. D. Tygar, Victor Wen, and David E. Culler. SPINS: security protocols for sensor networks. *Wireless Networks*, 8(5):521–534, September 2002.

[95] Joseph Polastre, Jason Hill, and David Culler. Versatile Low Power Media Access for Wireless Sensor Networks. In *Proc. 2nd ACM Int. Conf. on Embedded Networked Sensor Syst. (SenSys)*, pages 95–107. ACM, 2004.

[96] Ying Qiu, Shining Li, Dongyu Yang, and Zhigang Li. RWB: An Efficient Receiver-Initiated Single-Hop Broadcast Protocol for Asynchronous MAC in Wireless Sensor Networks. *Recent Advances in Comput. Sci. and Inform. Eng.*, 127:261–266, 2012.

[97] Qualcomm Atheros. WLAN: AR4100. `http://www.qca.qualcomm.com/technology/technology.php?nav1=47&product=114`.

[98] V. Raghunathan, A. Kansal, J. Hsu, J. Friedman, and M.B. Srivastava. Design considerations for solar energy harvesting wireless embedded systems. In *Proc. 4th Int. Symp. on Inform. Processing in Sensor Networks*, pages 457–462, 2005.

[99] Jérôme Rousselot, Amre El-Hoiydi, and Jean-Dominique Decotignie. WideMac: a Low Power and Routing Friendly MAC Protocol for Ultra Wideband Sensor Networks. In *Proc. IEEE Int. Conf. on Ultra-Wideband (ICUWB)*, volume 3, pages 105–108. IEEE, 2008.

[100] RTX A/S. Low Power Wi-Fi RTX41xx. `http://www.rtx.dk/LPW/RTX4100`.

[101] RTX A/S. RTX4100 Wi-Fi Module: Application Note AN1 Current Consumption. `http://www.rtx.dk/LPW/RTX4100`.

[102] W.K.-G. Seah, Zhi Ang Eu, and H. Tan. Wireless sensor networks powered by ambient energy harvesting (WSN-HEAP) - Survey and challenges. In *Proc. 1st Int. Conf. on Wireless Commun., Vehicular Technology, Inform. Theory and Aerospace Electronic Syst. Technology (VITAE)*, pages 1–5, 2009.

[103] Semtech. SX1212 Ultra-Low Power Integrated 300-510MHz Transceiver. `http://www.semtech.com/apps/filedown/down.php?file=sx1212.pdf`, 2009.

[104] Hui Song, Sencun Zhu, and Guohong Cao. In *Int. Conf. on Mobile Adhoc and Sensor Syst., title=Attack-resilient time synchronization for wireless sensor networks, year=2005, pages=8 pp.-772,.*

[105] William Stallings. *Cryptography and Network Security*. Prentice Hall, 2005.

[106] Yanjun Sun, Omer Gurewitz, Shu Du, Lei Tang, and David B. Johnson. ADB: An Efficient Multihop Broadcast Protocol based on Asynchronous Duty-cycling in Wireless Sensor Networks. In *Proc. 7th ACM Int. Conf. on Embedded Networked Sensor Syst. (SenSys)*, pages 43–56. ACM, 2009.

[107] Yanjun Sun, Omer Gurewitz, and David B. Johnson. RI-MAC: A Receiver-Initiated Asynchronous Duty Cycle MAC Protocol for Dynamic Traffic Loads in Wireless Sensor Networks. In *Proc. 6th ACM Int. Conf. on Embedded Networked Sensor Syst. (SenSys)*, pages 1–14. ACM, 2008.

[108] Bharath Sundararaman, Ugo Buy, and Ajay D. Kshemkalyani. Clock synchronization for wireless sensor networks: A Survey. *Ad Hoc Networks (Elsevier)*, 3:281–323, 2005.

[109] R. Sundaresan and S. Verdu. Robust decoding for timing channels. *IEEE Trans. on Inform. Theory*, 46(2):405–419, 2000.

[110] H. Tan, P.W.Q. Lee, W.K.-G. Seah, and Zhi Ang Eu. Impact of Power Control in Wireless Sensor Networks Powered by Ambient Energy Harvesting (WSN-HEAP) for Railroad Health Monitoring. In *Proc. Int. Conf. on Advanced Inform. Networking and Applicat. Workshops (WAINA)*, pages 804–809, 2009.

[111] Hong-wei Tang, Jian-nong Cao, Cai-xia Sun, and Kai Lu. REA-MAC: A low latency routing-enhanced asynchronous duty-cycle MAC protocol for wireless sensor networks. *J. of Central South University*, 20(3):678–687, 2013.

[112] Lei Tang, Yanjun Sun, Omer Gurewitz, and David B. Johnson. EM-MAC: A Dynamic Multichannel Energy-Efficient MAC Protocol for Wireless Sensor Networks. In *Proc. 12th ACM Int. Symp. on Mobile Ad Hoc Networking and Computing (MobiHoc)*, page 23. ACM, 2011.

[113] Lei Tang, Yanjun Sun, Omer Gurewitz, and David B. Johnson. PW-MAC: An Energy-Efficient Predictive-Wakeup MAC Protocol for Wireless Sensor Networks. In *Proc. 30th Ann. Joint Conf. IEEE Comput. and Commun. Soc. (INFOCOM)*, pages 1305–1313. IEEE, 2011.

[114] Dhiren Tejani, Ali Mohammed A. H. Al-Kuwari, and Vidyasagar Potdar. Energy Conservation in a Smart Home. In *Proc. of the 5th Intl. Conf. on Digital Ecosystems and Technology (DEST)*. IEEE, 2011.

[115] Texas Instruments. eZ430-RF2500 Development Tool. SLAU227E, 2009. Available at `http://www.ti.com/lit/ug/slau227e/slau227e.pdf`.

[116] Texas Instruments. CC2500: Low-Cost Low-Power 2.4 GHz RF Transceiver, 2011. Available at `http://www.ti.com/lit/ds/symlink/cc2500.pdf`.

[117] Texas Instruments. MSP430x2xx Family User's Guide. SLAU144J, 2013. Available at `http://www.ti.com/lit/ug/slau144j/slau144j.pdf`.

[118] The Ohio State University NEST team. A Line in the Sand: A DARPA-NEST Field Experiment, 2003.

[119] Sameer Tilak, Nael B. Abu-Ghazaleh, and Wendi Heinzelman. A Taxonomy of Wireless Micro-Sensor Network Models. *ACM Mobile Computing And Commun. Review*, 6:28–36, 2002.

[120] Christopher M. Vigorito, Deepak Ganesan, and Andrew G. Barto. Adaptive Control of Duty Cycling in Energy-Harvesting Wireless Sensor Networks. In *Proc. 4th IEEE Conf. on Sensor, Mesh and Ad Hoc Commun. and Networks (SECON)*, pages 21–30. IEEE, 2007.

[121] Madava D. Vithanage, Xenofon Fafoutis, Claus Bo Andersen, and Nicola Dragoni. Medium Access Control for Thermal Energy Harvesting in Advanced Metering Infrastructures. In *Proc. of IEEE Eurocon*. IEEE, 2013.

[122] T. Voigt, A. Dunkels, J. Alonso, H. Ritter, and J. Schiller. Solar-aware clustering in wireless sensor networks. In *Proc. 9th Int. Symp. on Comput. and Commun. (ISCC)*, volume 1, pages 238–243, 2004.

[123] Thiemo Voigt, Hartmut Ritter, and Jochen Schiller. Solar-Aware Routing in Wireless Sensor Networks. In *Personal Wireless Commun.*, volume 2775 of *Lecture Notes in Computer Science*, pages 847–852. Springer Berlin Heidelberg, 2003.

[124] Takahiro Wada, I-Te Lin, and Iwao Sasase. Asynchronous Receiver-Initiated MAC Protocol with the Stair-Like Sleep in Wireless Sensor Networks. In *Proc. 22nd Ann. IEEE Int. Symp. on Personal Indoor and Mobile Radio Commun. (PIMRC)*, pages 854–858. IEEE, 2011.

[125] Qui Wang, Mark Hempstead, and Woodward Yang. A Realistic Power Consumption Model for Wireless Sensor Network Devices. In *Proc. 3rd Ann. IEEE Commun. Soc. Conf. on Sensor, Mesh and Ad Hoc Commun. and Networks (SECON)*, pages 286–295, 2006.

[126] Wenye Wang, Yi Xu, and Mohit Khanna. A survey on the communication architectures in smart grid. *Comput. Networks*, 2011.

[127] Xinguo Wang and Qian Zhang. Opportunistic Cooperation in Low Duty Cycle Wireless Sensor Networks. In *Proc. IEEE Int. Conf. on Commun. (ICC)*, pages 1–5. IEEE, 2010.

[128] Kamin Whitehouse, Alec Woo, Fred Jiang, Joseph Polastre, and David Culler. Exploiting The Capture Effect For Collision Detection And Recovery. In *Proc 2nd IEEE Workshop on Embedded Networked Sensors (EmNetS-II)*, pages 45–52. IEEE, 2005.

[129] WindowMaster. NV Comfort. `http://www.windowmaster.com/Solutions/NV-Comfort%E2%84%A2.aspx`.

[130] Poonam Yadav and Julie A. McCann. YA-MAC: Handling Unified Unicast and Broadcast Traffic in Multi-hop Wireless Sensor Networks. In *Proc. 7th IEEE Int. Conf. and Workshops on Distributed Computing in Sensor Syst. (DCOSS)*, pages 1–9. IEEE, 2011.

[131] Dongyu Yang, Ying Qiu, Shining Li, and Zhigang Li. RW-MAC: An asynchronous receiver-initiated ultra low power MAC protocol for Wireless Sensor Networks. In *Proc. IET Int. Conf. on Wireless Sensor Network (IET-WSN)*, pages 393–398. IET, 2010.

[132] Wei Ye, John Heidemann, and Deborah Estrin. An Energy-Efficient MAC Protocol for Wireless Sensor Networks. In *Proc. 21st Ann. Joint Conf. IEEE Comput. and Commun. Soc. (INFOCOM)*, volume 3, pages 1567–1576. IEEE, 2002.

[133] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. Wireless sensor network survey. *Comput. Netw.*, 52(12):2292–2330, August 2008.

[134] Yueh-Tiam Yong, Chee-Onn Chow, Jeevan Kanesan, and Hiroshi Ishii. EE-RI-MAC: An energy-efficient receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks. *International Journal of Physical Sciences*, 6(11):2633–2643, 2011.

[135] Kai Zeng, Kui Ren, Wenjing Lou, and Patrick J. Moran. Energy Aware Efficient Geographic Routing in Lossy Wireless Sensor Networks with Environmental Energy Supply. *Wireless Networks*, 15(1):39–51, 2009.